TRABAJO FINAL DE MÀSTER

TÍTULO: DESARROLLO E INTEGRACIÓN DE UN SISTEMA CIBER-FÍSICO APLICADO AL CONTROL DE CALIDAD UTILIZANDO VISIÓN ARTIFICIAL.

AUTOR: JÁCOME RAMÍREZ, FERNANDO ISRAEL

FECHA DE PRESENTACIÓN: 19 de Diciembre, 2018

APELLIDOS: Jácome Ramírez **NOMBRE: Fernando Israel**

TITULACIÓN: MÁSTER UNIVERSITARIO EN INGENIERÍA DE SISTEMAS AUTOMÁTICOS Y ELECTRÓNICA INDUSTRIAL.

PLAN:

DIRECTOR: Jordi Prat Tasis

DEPARTAMENTO: EEL- INGENIERÍA ELECTRÓNICA.

CALIFICACIÓN DEL TFM

TRIBUNAL

PRESIDENTE **SECRETARIA** **VOCAL**

FECHA DE LECTURA:

Este proyecto tiene en cuenta aspectos medioambientales: € Sí € No

RESUMEN

Este proyecto pretende diseñar e implementar un prototipo de sistema ciber-físico como los utilizados en la industria. Dicho prototipo se enmarcará en el ámbito académico y estará enfocado al control de calidad de engranajes mediante la utilización de visión artificial.

El proyecto se dividirá en dos partes. El diseño y construcción de un sistema físico a escala que procesa mecánicamente los engranajes y la implementación de una aplicación web, que permitirá controlar, monitorizar y visualizar los resultados del procesamiento de los engranajes del sistema físico.

Para la consecución de estos objetivos, se ha construido un prototipo conformado por un sistema de banda de transporte, sensores, actuadores y una cámara para implementar un sistema de clasificación de engranajes por visión, además del uso de dispositivos microprocesadores y microcontroladores de bajo coste.

Este proyecto se ha implementado sobre un único protocolo de comunicación TCP, conocido como Servidor - Cliente, gracias a la ayuda de un entorno de ejecución para documentos Javascript como lo es Node.js con sus respectivas librerías socket.io, facilitando el desarrollo de aplicaciones Websockets en el cliente y en el servidor.

Se ha desarrollado un aplicativo web bajo el entorno de desarrollo HTML5 y JavaScript, que permite el control y monitorización del sistema físico, además de obtener información de los resultados guardados en un historial de los engranajes que hayan sido procesados y sus respectivas características técnicas.

El resultado final ha sido la construcción de un sistema físico a escala y el desarrollo del software para el control y monitorización del procesado de engranajes, con este prototipo se ha trabajado en una guía práctica para que los alumnos puedan interactuar con el sistema ciber-físico en un entorno acotado. De este modo se pretende contribuir al aprendizaje de este concepto tecnológico desde un punto de vista eminentemente práctico.

Palabras claves (máximo 10):

CPS	IoT	JSON	TCP
UART	socket.io	Node js	ArduinoJson
API	Visión Artificial		

ABSTRACT

This project aims to design and implement a prototype of a cyber-physical system such as those used in the industry. This prototype will be framed in the academic field and will be focused on the quality control of gears using artificial vision.

The project will be divided into two parts. The design and construction of a physical system to scale that mechanically inspection processes the spur gears and the implementation of a web application, which will allow to control, monitor and visualize the results of the processing of physical system gears .

To achieve these objectives, a prototype has been built consisting of a conveyor belt system, sensors, actuators and a camera to implement a gear-by-vision classification system, in addition to the use of low-cost microprocessor and microcontroller devices.

This project has been implemented on a single communication protocol TCP, known as Server - Client, through an execution environment for Javascript documents such as Node.js with their respective socket.io libraries, facilitating the development of applications Websockets on the client and on the server.

A web application has been developed under the HTML5 and JavaScript development environment, allowing the control and monitoring of the physical system, in addition to obtain information from the results stored in a history of the gears that have been processed and their respective technical characteristics.

The result has been the construction of a physical system to scale and the development the software to control and monitor of gear processing, this prototype has worked on a practical guide for students to interact with the cyber-physical system in a limited environment. This way was intended to contribute to the learning of this technological concept from an eminently practical point of view.

Keywords (10 maximum):

CPS	IoT	JSON	TCP
UART	socket.io	Node js	ArduinoJson
API	Artificial Vision		

INDICE

CONTENIDO DE TABLAS.

CONTENIDO DE FIGURAS

GLOSARIO DE SIGNOS, SÍMBOLOS, ABREVIATURAS, ACRÓNIMOS Y TÉRMINOS.

CPS	Cyber-Physical Systems
IoT	Internet of Thing
TCP	Transmission Control Protocol
HTML	HyperText Markup Language (*lenguaje de marcas de hipertexto).*
CSS	Cascading Style Sheets (*Hojas de estilo en cascada*)
TIC	Tecnologías de información y comunicación.
UART	Universal Asynchronous Receiver-Transmitter
JSON	JavaScript Object Notation (Notación de objeto de JavaScript).
API	Application programming interface (interfaz de programación de aplicaciones)
DOM	Document Object Model (*Modelo de Objetos del Documento*).
OPENCV	Open Source Computer Vision Library
VNC	Virtual Network Computing (*Computación Virtual en Red*)
RFID	Radio Frequency IDentification (*Identificación por radiofrecuencia*)
JS	JavaScript
BSD	Berkeley Software Distribution
SECVIA	Selección de Engranajes Con VIsión Artificial.

INTRODUCCIÓN.

La Industria 4.0 está incorporando las tecnologías digitales a través de los conocidos "habilitadores digitales", cuyo objetivo principal es potenciar al máximo las posibilidades que la nueva industria ofrece gracias al IoT y los sistemas ciber-físicos.
El Big Data, la inteligencia artificial, el machine learning, las últimas tecnologías digitales de comunicación, algoritmos y robótica se unen para dar lugar a los sistemas ciber-físicos, que combinan sistemas físicos con software de última generación y que van a situarse en el epicentro de este cambio industrial. En esta línea, el IoT jugará un papel fundamental a la hora de catalizar el enlace entre todos estos elementos, que van a trabajar hacia objetivos comunes en sus respectivos sectores. [1]

Los CPS (sistemas ciber-físicos) integran capacidades de computación, almacenamiento y comunicación con capacidades de seguimiento y/o control de objetos en el mundo físico. Normalmente están conectados entre sí y en algunos casos disponen de capacidad de aprender y evolucionar.
En definitiva, el objetivo último de los CPS es mejorar la calidad de vida de los ciudadanos mediante la monitorización y el control del mundo físico en el que viven usando capacidades del mundo cibernético. [2]

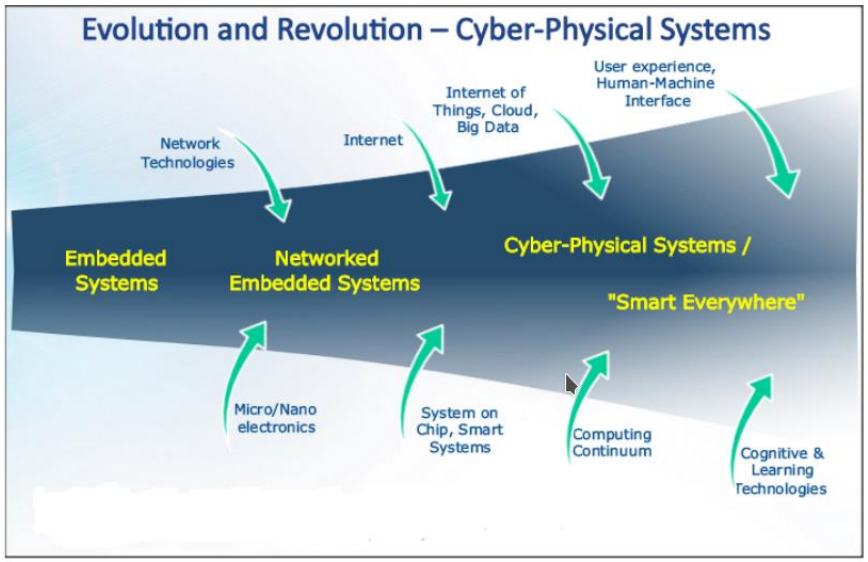


Figura 1. Evolución hacia los CPS, Fuente: [3]

Los CPS aportan una visión que va más allá de objetos individuales ofreciendo servicios a través de Internet. Podemos identificarlo con el concepto de Internet de las Cosas (IoT Internet of Things), ya que aborda sistemas completos que se componen dinámicamente a partir de otros sistemas y que son capaces de sentir y controlar el mundo físico, entendiéndolo y aprendiendo de las interacciones que se producen, de forma que evolucionan hacia la creación de entornos inteligentes.

Establecidas la importancia que hoy en día van adquiriendo los sistemas enfocados en el internet de las cosas dentro del ámbito industrial, este trabajo desarrolla conceptos puntuales para comprender, desarrollar e implementar dentro de un sistema a nivel académico un sistema Ciber-físico, con el fin de entender como dispositivos físicos se integran a capacidades computacionales y todo ello con dispositivos al alcance de un estudiante, como son el Raspberry pi y el Arduino.

Para ello se desarrollaró un prototipo SECVIA (Selección de Engranajes Con Visión Artificial). SECVIA es una plataforma de aprendizaje mediante sistema de banda transportadora a escala y enfocado al control de calidad mediante la utilización de visión artificial. En la industria dentro del área de producción, su función principal es el control de calidad, asegurando que los productos cumplan con los requerimientos mínimos predefinidos de calidad. Este prototipo tiene como objetivo verificar si un engrane cuenta con los dientes correspondientes, así como datos técnicos importantes como: Diámetro exterior (De), Diámetro interior (Di), Número de dientes (z), Módulo(M) y paso (P) de los dientes.

Con este análisis de control de calidad el sistema interpreta si el engranaje cuenta con los dientes completos o bien tiene pérdida de dientes, reflejando como resultados los dientes perdidos y la posición en el engranaje de los respectivos dientes defectuosos.

1. OBJETIVOS

El objetivo principal de este trabajo final de máster es la creación de un entorno ciber físico de nivel académico, que permita la ejecución de cierto sistema físico desde un sitio web de internet. La aplicación ayudará a aplicar las distintas tecnologías que conforman los sistemas cíber-físicos en la industria.
Con el entorno Ciber-físico a desarrollar se intentará comprender aspectos relacionados de este concepto, logrando avanzar en el conocimiento de las IoT. Ello será posible mediante la posibilidad de:

- Integrar sensores y actuadores dentro de un sistema ciber-físico a escala.
- Desarrollar un interfaz de comunicación de los diferentes dispositivos del aplicativo.
- Ser capaz de monitorear, coordinar y controlar en tiempo real desde cualquier punto de la red las funciones integradas por un núcleo de procesamiento y comunicación.
- Procesar y clasificar engranajes en función de su calidad a través del sistema de visión.

Como conclusión, este trabajo contará con dos aspectos importantes para el aprendizaje de un CPS:

- Diseño y montaje mecánico y electrónico de un sistema físico a escala.
- Desarrollo de software (algoritmos) para el control y monitorización del procesamiento de objetos en el ámbito de control de calidad de una empresa.

2. ESTADO DEL ARTE

En la actualidad los sistemas ciber-físicos van formando parte de las aplicaciones relacionadas con el internet de las cosas, gracias a la integración sistemas computacionales. En el mercado industrial, las infraestructuras de interconexiones de los CPS entre objetos, personas, sistemas físicos y fuentes de información están permitiendo el procesamiento de información de manera inteligente hacia un mundo virtual.

Si nos ubicamos dentro del ámbito industrial, los CPS están siendo integrados con la finalidad de lograr contar con sistemas autónomos y con la posibilidad de evolucionar, llegando a ser independientes en toma de decisiones específicas para la cual son diseñadas. En vista a que estos sistemas trabajan de forma autónoma, la necesidad de un sistema virtual y en tiempo real para controlar y monitorizar desde puntos indistintos de internet, permitirá verificar su respectiva funcionalidad. Podemos encontrar distintos campos de aplicación tal y como se muestra en la Tabla 1.

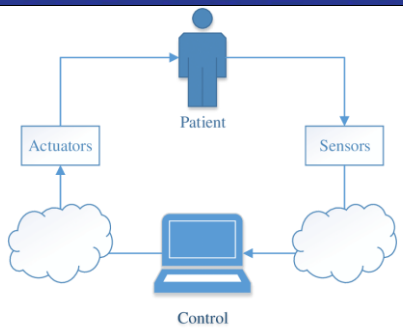
Área de aplicación	Ventajas	Ilustración del campo de aplicación
Transporte y logística	Aviones más rápidos y seguros. Uso mejorado del tráfico aéreo. Autos más seguros y eficientes.	
Salud y Biomedicina	Incrementos de usos efectivos de cuidados en casa. Dispositivos competentes para un fácil diagnóstico. Desarrollos de nuevas prótesis internas y externas.	
Infraestructura	Redes eléctricas con mayor confiabilidad Carreteras que permiten un tráfico con mayor la seguridad	
Energía y Automatización Industrial	Eficiencia energética y más barato para operar	

Tabla 1. Aplicaciones industriales de los CPS

Si nos ubicamos en el ámbito académico, después de la investigación realizada, no se han encontrado proyectos enfocados al estudio práctico de los CPS. Por otra parte, las fuentes que se ha resaltado evitan diferenciar claramente entre IoT y CPS.

Los proyectos académicos desarrollados están centrados en su mayor parte hacia sistemas conocidos como Smart City, Domótica y almacenamiento de datos, y estos enfocados al control de iluminación, prototipos de riegos automáticos, clasificadores de objetos en función del material. A continuación, se describen estos limitados prototipos dentro del campo académico.

- Smart City (Prototipos a escala de una ciudad inteligente).
  Diferentes centros académicos de proyectos se encuentran desarrollando prototipos de ciudades para la implementación de redes inteligentes, cuyas funciones son recolectar energía solar, proporcionar servicios de señalética, sensores conectados para tareas de detección, control de tráfico, etc.

Figura 2. Smart City de Tejas Dubhir [4]

- Domótica con FIREBASE. Google en el año 2014 adquirió una firebase para desarrollar proyectos relacionados con internet de las cosas, además de contar una plataforma para desarrollar aplicaciones web y aplicaciones móviles, plataforma que está destinada al análisis, almacenamiento de datos, notificaciones y reportes. En la Figura 3, se muestra una aplicación enfocada a la domótica controlada desde una aplicación web.

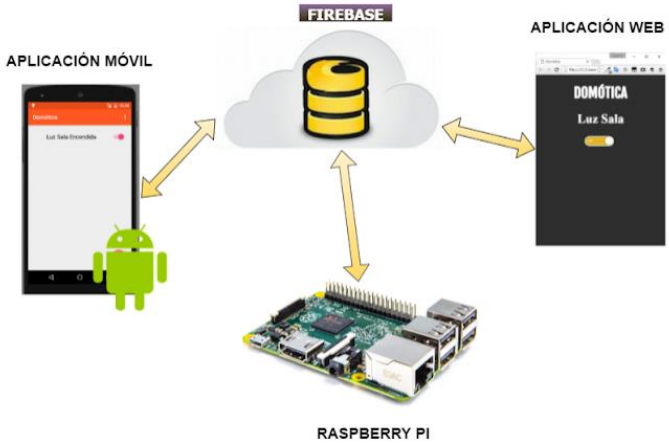


Figura 3. Domótica con Firebase Google. [5]

- Estación de trabajo didáctica de un sistema de clasificación de objetos por color de la empresa FESTO.
  FESTO es una empresa del ámbito industrial de sistemas de control industrial y automatización. Su producción está relacionada con la tecnología de accionamiento y control automático y eléctrico. Además de contar con estaciones didácticas de aprendizaje en sistemas orientado en la industria, uno de ellos es la clasificación de

objetos a partir de un sensor de color. Sin embargo, la estación no cuenta con tecnología de visión y aplicativos webs para su control y monitorización.
En la Figura 4, se muestra la estación de clasificación de Festo controlada por un autómata programable y electroválvulas para accionamiento de los distintos actuadores.

Figura 4. Centro de clasificación por color didáctico de Festo [6]

Los sistemas ciber-físicos se enfocan en introducir aplicaciones web y a la vez sistemas computacionales en una plataforma física. No se ha encontrado en el mercado prototipos de aprendizaje en el ámbito industrial.
Este trabajo intenta integrar en una plataforma física a escala un sistema ciber-físico mediante el desarrollo de un aplicativo web. El sistema se encargará del procesamiento de engranajes rectos a partir de una captura por un dispositivo con tecnologías de visión, y será capaz de decidir el estado del engranaje procesado.

3. MARCO TEÓRICO

3.1 Internet de las Cosas (IOT).

El término “Internet de las Cosas” (IoT) fue empleado por primera vez en 1999 por el pionero británico Kevin Ashton para describir un sistema en el cual los objetos del mundo físico se podían conectar a Internet por medio de sensores.
Ashton acuñó este término para ilustrar el poder de conectar a Internet las etiquetas de identificación por radiofrecuencia (RFID) que se utilizaban en las cadenas de suministro corporativas para contar y realizar un seguimiento de las mercancías sin necesidad de intervención humana. Hoy en día, el término Internet de las Cosas se ha popularizado para describir escenarios en los que la conectividad a Internet y la capacidad de cómputo se extienden a una variedad de objetos, dispositivos, sensores y artículos de uso diario. [7]

El «Internet de las Cosas» (IoT) hace referencia, como se ha adelantado, a una tecnología basada en la conexión de objetos cotidianos a Internet que intercambian, agregan y procesan información sobre su entorno físico para proporcionar servicios de valor añadido a los usuarios finales. También reconoce eventos o cambios, y tales sistemas pueden reaccionar de forma autónoma y adecuada. Su finalidad es, por tanto, brindar una infraestructura que supere la barrera entre los objetos en el mundo físico y su representación en los sistemas de información.
Esta integración de sensores y dispositivos en objetos cotidianos que quedan conectados a Internet a través de redes alámbricas e inalámbricas ha alumbrado, un nuevo modo de interacción en el mundo físico, inspirado en la idea de ubicuidad y facilitado por el desarrollo de las TIC y la industria electrónica. [8]

3.2 Sistemas Ciber-Físicos.

El término Cyber-Physical System o CPS cuenta con diferentes significados, ciertas veces se usa como un sinónimo de sistema en el que la computación interactúa con el mundo físico, otras como sistema empotrado en red y en otras ocasiones como sistema de sistemas.
La fundación americana NSF (National Science Foundation) describe los CPS como sistemas construidos a partir de la integración transparente de componentes físicos y computacionales, que permitirán superar a los simples sistemas empotrados actuales en cuanto a capacidad, adaptabilidad, escalabilidad, resiliencia, seguridad y usabilidad. [9]

El programa europeo Horizonte 2020 en su topic *H2020-ICT-01-2014* (Smart Cyber-Physical Systems) indica que los CPS son la próxima generación de sistemas TIC empotrados que se interconectan y colaboran a través del Internet de las Cosas, y proporcionan a los ciudadanos y a los negocios un conjunto amplio de aplicaciones y servicios innovadores. Son los sistemas TIC que están empotrados cada vez en más tipos de objetos, dotándolos de inteligencia y de mayor eficiencia energética y facilidad de uso, por ejemplo, en los sistemas de transporte, los coches, las fábricas, los hospitales, las oficinas, los hogares, las ciudades y los dispositivos personales. [10]

Por su parte, la iniciativa tecnológica conjunta ECSEL en su plan estratégico multianual [1] considera los CPS como una de las capacidades tecnológicas esenciales para dotar de inteligencia a todos los sectores y ámbitos (movilidad, sociedad, energía, salud, producción), junto a la micro/nano electrónica y la integración de sistemas. En concreto, los CPS se consideran como la próxima generación de sistemas TIC empotrados inteligentes interconectados, interdependientes, colaborativos y autónomos, que proporcionan

computación y comunicación, así como monitorización y/o control de componentes/procesos físicos en diferentes dominios de aplicación, incluido los de seguridad crítica. [11]

En concreto, los CPS son sistemas realimentados que están en red y/o distribuidos a través de sensorización y actuación inalámbrica, que son adaptativos, predictivos, inteligentes y en tiempo real.

En la Figura 5 se muestra una visión general sobre el concepto y los campos de aplicación, es decir, son sistemas retroalimentados con la posibilidad de insertarse dentro de ámbitos económicos y humanos, con los debidos requerimientos de seguridad, diseño de herramientas y metodologías que soporten características de validación análisis modelado de los sistemas. Así los CPS tienen aplicaciones en gran parte del campo industrial.

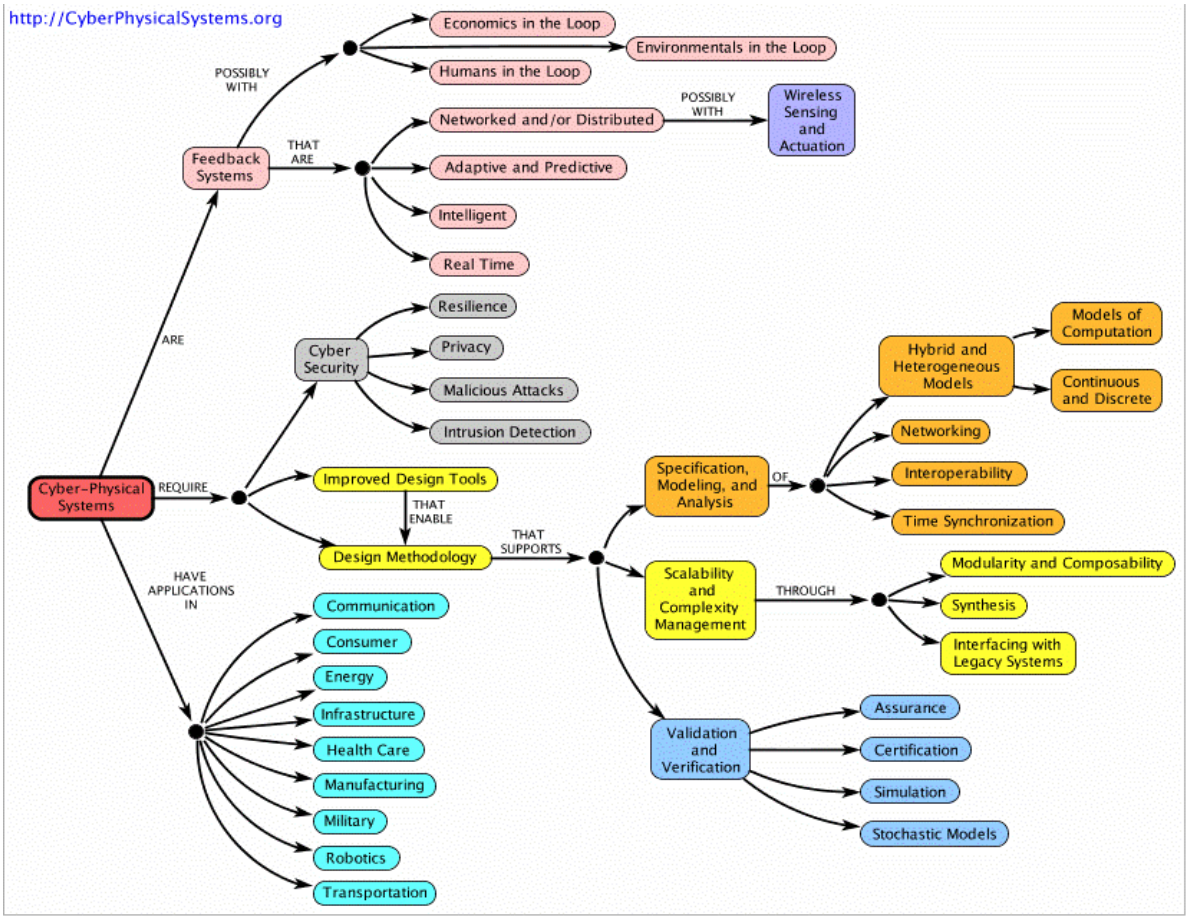


Figura 5. Mapa conceptual de los CPS, Fuente: [12]

Para resumir y diferenciar entre el IoT y los CPS, a continuación, se describe los conceptos de cada uno de estos de manera concisa.

IoT, es una tecnología basado en la comunicación que permite conectar “cosas” y tienen la capacidad de identificar, sensar y procesar, y la vez interactuar entre máquinas, personas y objetos dentro de un entorno.

Por otro lado, los CPS son sistemas diseñados que integran funciones de detección, algoritmos computacionales, control y redes en objetos físicos e infraestructura, que permiten superar a los simples sistemas empotrados actuales en cuanto a capacidad, adaptabilidad, escalabilidad, resiliencia, seguridad y usabilidad.

Establecido el concepto de CPS nuestro objetivo será el desarrollo de un prototipo a escala con fines educativos, que permita entender qué es un CPS, cuál es su utilidad en la industria y que ventajas es capaz de aportar.

En la Figura 6 se presenta la arquitectura del CPS académico a implementar, es considerado al microprocesador como dispositivo principal, cuya funcionalidad es contener al servidor Web y realizar la interconexión entre el aplicativo web y el prototipo, mediante la estructura JSON, del procesamiento de una imagen (engrane) capturada desde un dispositivo de visión (cámara). Y, por último, la utilización de la comunicación UART con la plataforma Arduino, este enviará órdenes a los actuadores y recibirá respuestas del estado de los sensores del sistema a implementar.

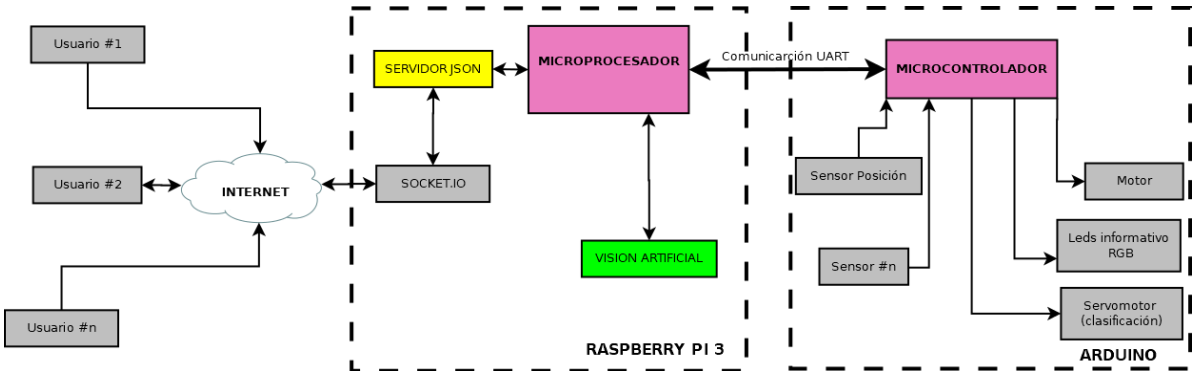


Figura 6. Arquitectura CPS a implementar.

Para el desarrollo e implementación de un CPS académico, primeramente, se deben conocer los conceptos básicos para la creación de un servidor desde el Raspberry, así como la comunicación entre este servidor – cliente y dispositivos físicos, y por otra parte contar con sistema computacional implementado a una cámara de visión artificial basado en las librerías de Opencv.

3.3 Cliente-Servidor

TCP (Transmission Control Protocol) es un protocolo orientado a conexión. No hay relaciones maestro/esclavo. Las aplicaciones, sin embargo, utilizan un modelo cliente/servidor en las comunicaciones.

Un servidor es una aplicación que ofrece un servicio a usuarios de Internet; un cliente es el que pide ese servicio. Una aplicación consta de una parte de servidor y una de cliente, que se pueden ejecutar en el mismo o en diferentes sistemas.

Los usuarios invocan la parte cliente de la aplicación, que construye una solicitud para ese servicio y se la envía al servidor de la aplicación que usa TCP/IP como transporte.

El servidor es un programa que recibe una solicitud, realiza el servicio requerido y devuelve los resultados en forma de una respuesta. Generalmente un servidor puede tratar múltiples peticiones (múltiples clientes) al mismo tiempo. [13].

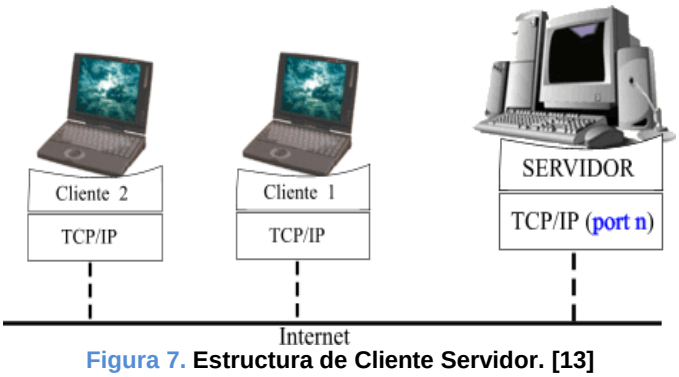


Figura 7. Estructura de Cliente Servidor. [13]

- ## Utilidad de aplicaciones de un Cliente-Servidor

Una aplicación cliente/servidor, es un programa que cuenta con un front-end que establece una conexión directa a través de una red, con un servidor que aloja el programa, servicio o desarrollo informático al cual deseamos acceder.

Directamente desde el terminal y en tiempo real, envía toda la información suministrada por el usuario al servidor en donde se ejecuta la acción y el mismo servidor regresa el resultado a la máquina cliente, desde donde se ha ingresado los datos, mostrando por pantalla toda la información que ha sido procesada [14].

Entre las principales utilidades que le podemos asignar a las aplicaciones cliente/servidor destacan las siguientes:

- Ligereza impresionante, ya que toda la carga y consumo de recursos se hace directamente en el servidor.
- Facilidad de mantenimiento, ya que se hace mantenimiento general en el código principal en el servidor.
- Posibilidad de realizar trabajos dinámicos y en grupo.
- Servicio de trabajo incluso en zonas remotas, gracias a Internet.
- Posibilidad de ser desarrolladas en diversos lenguajes de programación.
- Capacidad de trabajar, por lo general, en diferentes plataformas de manera simultánea.

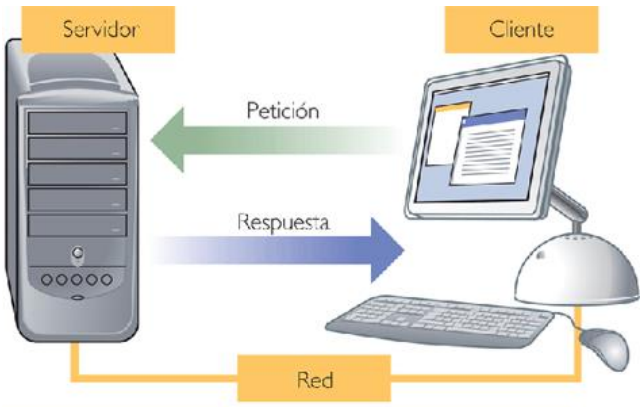


Figura 8. Comunicación servidor - cliente [14]

Establecido el concepto y la utilidad de las aplicaciones Servidor-cliente, se deben conocer los entornos de desarrollo a utilizar en el prototipo de aplicación, así como sus herramientas y bibliotecas.

3.3.1 WebSocket

WebSocket es una tecnología que proporciona un canal de comunicación bidireccional y full-duplex sobre un único socket TCP. Está diseñada para ser implementada en navegadores y servidores web, pero puede ser utilizado por cualquier aplicación cliente/servidor.

La especificación HTML5 WebSockets define una API que permite a las páginas web utilizar el protocolo WebSockets para la comunicación bidireccional con un host remoto. Introduce la interfaz WebSocket y define un canal de comunicación dúplex completo que opera a través de un único socket a través de la Web. Los WebSockets de HTML5 proporcionan una reducción enorme en el tráfico de red innecesario y la latencia en comparación con las soluciones de sondeo ineludible y de sondeo largo que se usaron para simular una conexión de dúplex completo mediante el mantenimiento de dos conexiones.

Los WebSockets de HTML5 tienen en cuenta riesgos de red como proxies y firewalls, posibilitan la transmisión a través de cualquier conexión, y con la capacidad de admitir comunicaciones ascendentes y descendentes en una sola conexión, las aplicaciones basadas en WebSockets HTML5 colocan menos carga en los servidores, permitiendo que las máquinas existentes sean compatibles más conexiones concurrentes. Una arquitectura básica basada en WebSocket en la cual los navegadores usan una conexión WebSocket para comunicación directa full-duplex con hosts remotos. [15]

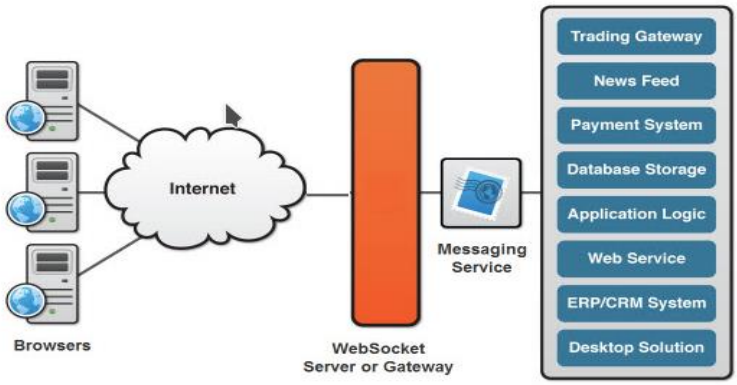


Figura 9. HTML5 usando WebSocket, [16]

Las aplicaciones estilo cometa generalmente emplean largas encuestas como una línea de defensa rudimentaria contra firewalls y servidores proxy. La técnica es efectiva, pero no es adecuada para aplicaciones que tienen una latencia inferior a 500 milisegundos o requisitos de alto rendimiento. Las tecnologías basadas en complementos, como Adobe Flash, también proporcionan cierto nivel de soporte de socket, pero durante mucho tiempo han tenido que soportar los problemas de proxy y cortafuegos que WebSockets resuelve ahora.

WebSockets, como otras piezas del esfuerzo de HTML5, como Almacenamiento local y Geolocalización, originalmente formaba parte de la especificación HTML5, pero se trasladó a un documento de estándares separado para mantener la especificación enfocada. WebSockets ha sido enviado al Grupo de trabajo de ingeniería de Internet (IETF) por sus creadores, el Grupo de trabajo de tecnología de aplicaciones de hipertexto web (WHATWG). Los autores, los evangelistas y las compañías involucradas en la estandarización todavía se refieren al conjunto original de características, incluyendo WebSockets, como "HTML5". [16]

- ## Tecnología node js

Node.js es un entorno en tiempo de ejecución multiplataforma, de código abierto, para la capa del servidor (pero no limitándose a ello) basado en el lenguaje de programación ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos y basado en el motor V8 de Google. Fue creado con el enfoque de ser útil en la creación de programas de red altamente escalables, como por ejemplo, servidores web. Fue creado por Ryan Dahl en 2009 y su evolución está apadrinada por la empresa Joyent, que además tiene contratado a Dahl en plantilla

Node.js funciona con un modelo de evaluación de un único hilo de ejecución, usando entradas y salidas asíncronas las cuales pueden ejecutarse concurrentemente en un número de hasta cientos de miles sin incurrir en costos asociados al cambio de contexto. Este diseño de compartir un único hilo de ejecución entre todas las solicitudes atiende a necesidades de aplicaciones altamente concurrentes, en el que toda operación que realice entradas y salidas debe tener una función callback. Un inconveniente de este enfoque de único hilo de ejecución es que Node.js requiere de módulos adicionales como cluster para escalar la aplicación con el número de núcleos de procesamiento de la máquina en la que se ejecuta. [17]

- ## Biblioteca SOCKET.IO

Socket.IO es una biblioteca de JavaScript para aplicaciones web en tiempo real. Permite la comunicación bidireccional en tiempo real entre clientes web y servidores. Tiene dos partes: una biblioteca del lado del cliente que se ejecuta en el navegador y una biblioteca del lado del servidor para Node.js. Ambos componentes tienen una API casi idéntica. Al igual que Node.js, está orientado a eventos. [18].

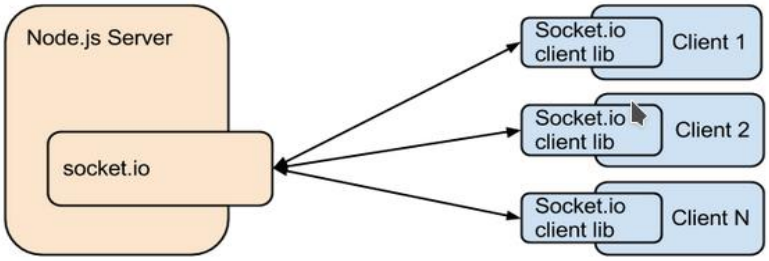


Figura 10. Función de SOCKET.IO [19]

Socket.IO utiliza principalmente el protocolo WebSocket con sondeo como opción alternativa, a la vez que proporciona la misma interfaz. Aunque se puede usar simplemente como un contenedor para WebSocket, ofrece muchas más características, incluida la transmisión a múltiples sockets, el almacenamiento de datos asociados con cada cliente y las E / S asíncronas.

Socket.IO proporciona la capacidad de implementar análisis en tiempo real, transmisión binaria, mensajería instantánea y colaboración de documentos. Los usuarios notables incluyen Microsoft Office, Yammer y Zendesk.
Socket.IO maneja la conexión de forma transparente. Se actualizará automáticamente a WebSocket si es posible. Esto requiere que el programador solo tenga conocimiento de Socket.IO. [18].

- ## Node.js + Socket.io = Real-Time IO

Definitivamente Node.js está ganando relevancia como una de las tecnologías más populares en esta área. Hay varias razones que explican esta tendencia, pero claramente uno de ellos es el modelo asincrónico orientado a eventos y sus ventajas para lidiar con problemas de BigData.

Cuando se trata de requisitos en tiempo real, Socket.io puede jugar un papel importante en la arquitectura de la aplicación web, proporcionando una capa de abstracción para la comunicación entre el navegador y el servidor. El modelo impulsado por eventos Node.js combinado con las capacidades en tiempo real de Socket.io ofrece una buena solución para enfrentar los desafíos de BigData en dominios donde la capacidad en tiempo real es importante. [20]

3.3.2 Sitio Web

La World Wide Web o simplemente WWW o Web es uno de los métodos más importantes de comunicación que existe en Internet. Consiste en un sistema de información basado en Hipertexto (texto que contiene enlaces a otras secciones del documento o a otros documentos). La información reside en forma de páginas Web en ordenadores que se denominan servidores Web y que forman los nodos de esta telaraña.

La Web tiene una estructura Cliente-Servidor de forma que los proveedores de información (servidores) atienden a las peticiones de los programas cliente (normalmente denominados navegadores o browsers) que son manejados por el usuario final. Además, este software cliente, por ser más moderno, tiene la peculiaridad de que es capaz de entenderse con otro tipo de servicios o protocolos, como son ftp, news, gopher..., de manera que sólo hace falta un programa para utilizar todos los servicios disponibles en la red. [21]

Un sitio web es una estructura de información y/o comunicación generada en el nuevo ámbito o espacio de comunicación (Internet), creado por la aplicación de las tecnologías de la información (tecnologías de creación, mantenimiento y desarrollo de los sitios web), que posee dos elementos fundamentales (acciones de los sujetos y contenidos) y en donde se plantean un conjunto de prestaciones que los usuarios que visitan dicho web pueden ejercitar para satisfacer una o varias necesidades que posean.

Los dos elementos fundamentales de un sitio web son las acciones de los sujetos y los contenidos, pero especificando un poco más diremos que los contenidos y las acciones que los sujetos realizan con aquellos, así como con otros sujetos. Hablar de las acciones de los sujetos, así como de los contenidos, es más que relevante, pues identificamos los dos principales elementos vertebrales del web. [22]

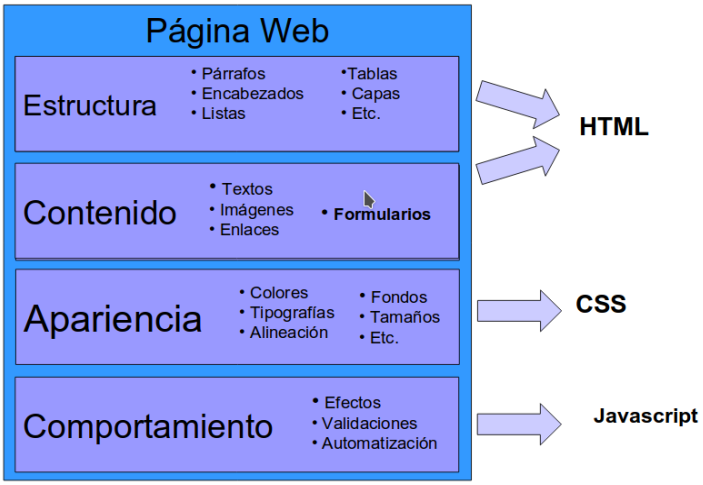


Figura 11. Estructura de una página WEB [23]

- **HTML5(Hyper Text Markup Language).**

Es el elemento de construcción más básico de una página web y se usa para crear y representar visualmente una página web. Determina el contenido de la página web, pero no su funcionalidad. Otras tecnologías distintas de HTML son usadas generalmente para describir la apariencia/presentación de una página web (CSS) o su funcionalidad (JavaScript).

HTML le da "valor añadido" a un texto estándar en español. Hiper Texto se refiere a enlaces que conectan una página Web con otra, ya sea dentro de una página web o entre diferentes sitios web. los vínculos son un aspecto fundamental de la Web. Al subir contenido a Internet y vincularlo a páginas de otras personas, te haces participante activo de esta Red Mundial. [24]

Figura 12. Estructura de diseño de texto una página web

- **CSS (Cascading Style Sheets)**

Hojas de Estilo en Cascada es el lenguaje utilizado para describir la presentación de documentos HTML o XML, esto incluye varios lenguajes basados en XML como son XHTML o SVG. CSS describe como debe ser renderizado el elemento estructurado en pantalla, en papel, hablado o en otros medios.

Los navegadores Web, al aplicar las reglas CSS a un documento, modifican la manera en que este es presentado. Una regla CSS se compone de:

- Un conjunto de propiedades, con valores establecidos para actualizar la presentación del contenido HTML, por ejemplo, quiero que el ancho de un elemento sea el 50% de su elemento padre, y que su fondo sea rojo.
- Un selector, que seleccionará los elementos afectados por el nuevo valor de la propiedad.

El conjunto de reglas CSS contenidas en el documento de estilos (stylesheet) afectará a la presentación de la página web. Profundizaremos en la sintaxis CSS en el siguiente artículo del módulo — Sintaxis CSS. [25]

Cuando un navegador muestra un documento, debe combinar su contenido con su información de estilos. Procesa el documento en dos fases:

1. El navegador convierte **HTML** y **CSS** en un **DOM** (*Objeto Documento Modelo*). Este DOM representa el documento en la memoria del ordenador. Combinando el contenido del documento con su estilo.
2. El navegador muestra el contenido del DOM.

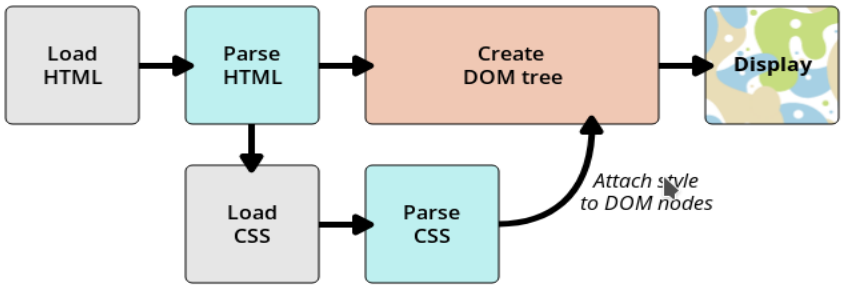


Figura 13. **Comportamiento HTML con CSS. [25]**

- **JAVASCRIPT**

JavaScript es un lenguaje de programación que te permite realizar actividades complejas en una página web, cada vez más una página web hace más cosas que sólo mostrar información estática como mostrar actualizaciones de contenido en el momento, interactuar con mapas, animaciones gráficas 2D/3D etc. Puedes estar seguro que JavaScript está involucrado. Es la tercera capa del pastel de los estándares en las tecnologías para la web, dos de las cuales son (HTML y CSS). [24]

JavaScript puede funcionar como lenguaje procedimental y como lenguaje orientado a objetos. Los objetos se crean programáticamente añadiendo métodos y propiedades a lo que de otra forma serían objetos vacíos **en tiempo de ejecución**, en contraposición a las definiciones sintácticas de clases comunes en los lenguajes compilados como C++ y Java. Una vez se ha construido un objeto, puede usarse como modelo (o prototipo) para crear objetos similares. [24]

3.3.3 JSON

JSON (JavaScript Object Notation - Notación de Objetos de JavaScript) es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, Standard ECMA-262 3rd Edition - Diciembre 1999. JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros. Estas propiedades hacen que JSON sea un lenguaje ideal para el intercambio de datos.
JSON está constituido por dos estructuras:

- Una colección de pares de nombre/valor. En varios lenguajes esto es conocido como un objeto, registro, estructura, diccionario, tabla hash, lista de claves o un arreglo asociativo.
- Una lista ordenada de valores. En la mayoría de los lenguajes, esto se implementa como arreglos, vectores, listas o secuencias.

Estas son estructuras universales; virtualmente todos los lenguajes de programación las soportan de una forma u otra. Es razonable que un formato de intercambio de datos que es independiente del lenguaje de programación se base en estas estructuras.
En JSON, se presentan de estas formas:

- Un *objeto* es un conjunto desordenado de pares nombre/valor. Un objeto comienza con { (*llave de apertura*) y termine con } (*llave de cierre*). Cada nombre es seguido por : (dos puntos) y los pares nombre/valor están separados por , (*coma*).
- Un *arreglo* es una colección de valores. Un arreglo comienza con [(corchete izquierdo) y termina con] (*corchete derecho*). Los valores se separan por , (*coma*).
- Un valor puede ser una cadena de caracteres con *comillas* dobles, o un *número*, o true o false o null, o un objeto o un arreglo. Estas estructuras pueden anidarse.
- Una *cadena de caracteres* es una colección de cero o más caracteres Unicode, encerrados entre comillas dobles, usando barras divisorias invertidas como escape. Un carácter está representado por una cadena de caracteres de un único carácter. Una *cadena de caracteres* es parecida a una cadena de caracteres C o Java.
- Un *número* es similar a un número C o Java, excepto que no se usan los formatos octales y hexadecimales.

Los espacios en blanco pueden insertarse entre cualquier par de símbolos. [26]

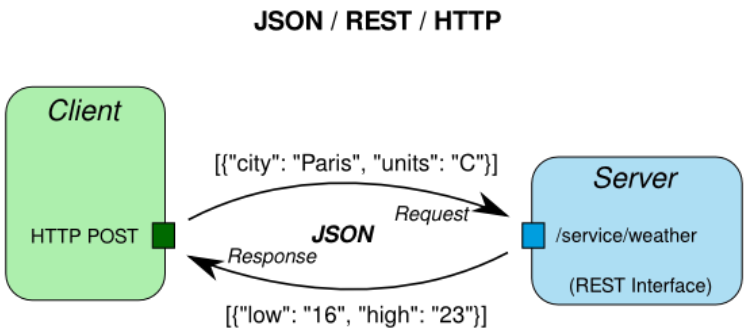


Figura 14. Solicitud/Respuesta con JSON. [27]

• LIBRERÍA ARDUINOJSON

Arduino facilita una clase llamada Strings, ofreciendo el uso de cadenas de caracteres con unos métodos muy sencillos de usar y poder usar los operadores que se conoce como Strings.

Clase que permite usar y manipular cadenas de texto de una forma más sencilla que los strings. Puedes concatenar, añadir, buscar, etc. usando los métodos/funciones que ofrece esta clase.

ArduinoJson es una libreria JSON C++ desarrollada para Arduino (serialización y deserialización en sistemas embebidos y trabajar en el entorno de la industria basado en el internet de las cosas, dentro de las características de ArduinoJson:

- Decodificar objetos JSON (soportan comentarios)

- Codificar objetos JSON (con identidad opcional)
- Elegante API (Application Programming Interface)
- Asignación de memoria fija, lo que permite trabajar en dispositivos con muy poca RAM.
- Sin duplicación de datos
- Autónomo (sin dependencia externa)
- Flujos de entrada y salida

[28]

3.4 OPENCV (Open Source Computer Vision Library).

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicaciones de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

Open CV es multiplataforma, existiendo versiones para GNU/Linux, Mac OS X y Windows. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica. [29]

- ### Procesamiento de imágenes

El desarrollo de algoritmos para el procesamiento de imágenes ha cobrado mayor impulso en el ámbito científico, en especial en el campo de la inteligencia artificial para aplicaciones relacionadas con visión por computadora. Este conocimiento se nutre de la digitalización de datos en dispositivos para captar videos, fotos, etcétera. Las imágenes, entonces, se convierten en el núcleo de la descripción de la realidad.

Entre los espacios de color más utilizados para el procesamiento de imágenes están el rojo, el verde y el azul (RGB). Según la Comisión Internacional de Iluminación, en su norma técnica CIE 1931-RGB, el color se conforma por tres dimensiones monocromáticas perceptibles al campo visual. Estos no son los únicos espacios de color; existen, entre otros, los espacios HSV (Hue-Saturation-Value) y los espacios L*a*b, más orientados a describir la percepción del color ante un cambio en alguna dimensión de la imagen. [30]

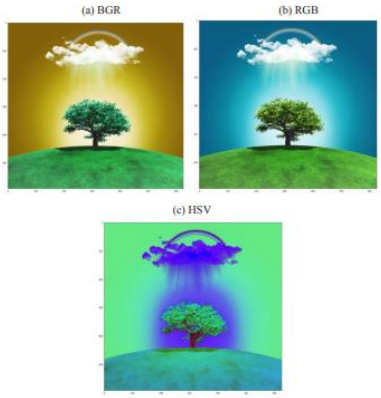


Figura 15. Espacios de color para el procesamiento de una imagen. [30]

4. DESARROLLO E IMPLEMENTACIÓN.

Este trabajo contará con dos partes importantes: el montaje de un sistema físico a escala y el desarrollo software y hardware para el control y procesamiento de engranes rectos.

Uno de los objetivos es comprender el comportamiento de un CPS, cuando el usuario interactúa con el sistema físico desde cualquier punto de la red. El trabajo se centrará dentro de una red de una organización (intranet).

4.1 Diseño mecánico del sistema Físico.

Enfocados en la aplicación a realizar, el prototipo a construir contará con un sistema de banda de transporte y una torre con una cámara que permitirá implementar un sistema de procesamiento de visión de la imagen. En la aplicación el producto u objeto a procesar son engranajes de dientes rectos. En la Figura 16 se muestra el diseño y ensamble de las partes mecánicas a construir.

El prototipo SECVIA incorpora por un motor de DC a 12 Voltios, para otorgar el correspondiente movimiento de la cinta de transporte de engranajes. Además, cuenta con sensores de posición que facilitarán la detección del engranaje en proceso. Cuando el engranaje haya sido detectado y ubicado mediante una cámara con tecnología de visión con su respectiva iluminación, capturará la imagen para ser procesado. Con los resultados del procesamiento el uso de un motor de clasificación permitirá distinguir los engranes buenos y malos.

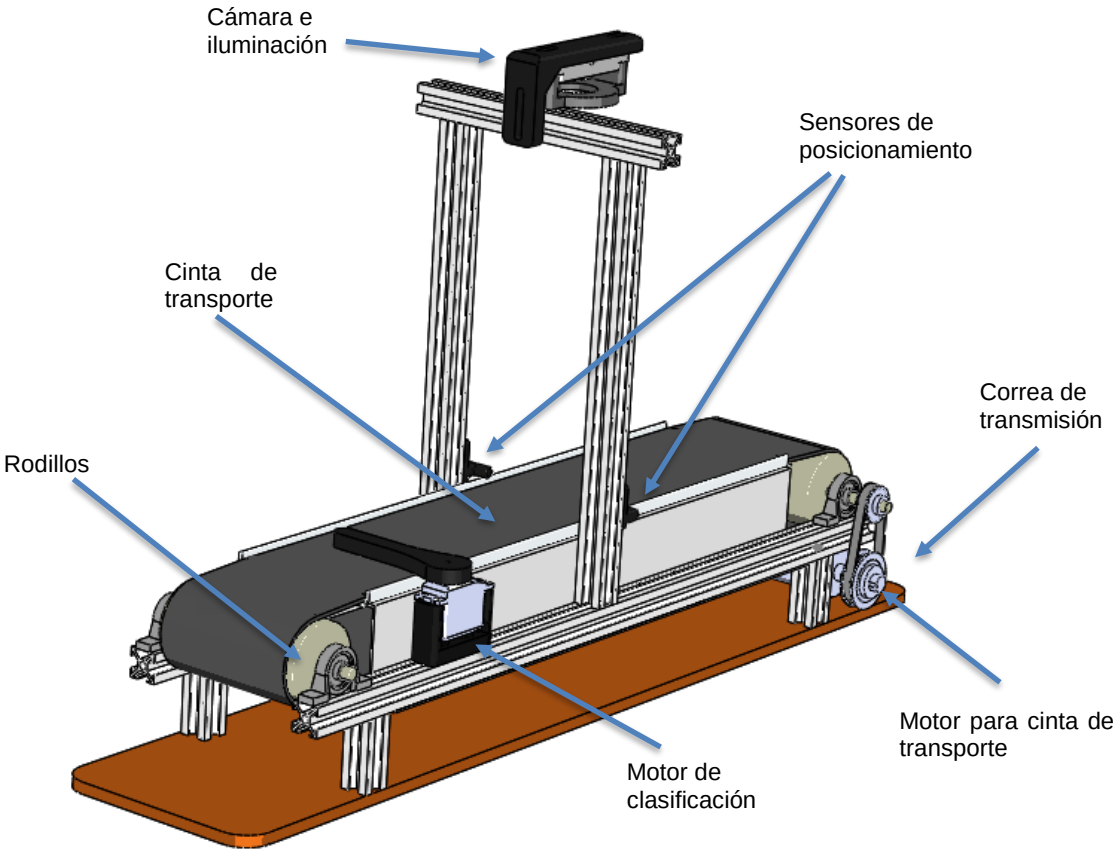


Figura 16. Diseño y ensamble del prototipo físico a desarrollar.

4.2 Diseño electrónico.

Para la estimulación del funcionamiento del prototipo SECVIA, se requiere de un diseño electrónico, diseño que involucra ramas fundamentales como los son: hardware y firmware. El hardware cumplirá funciones de obedecer las órdenes del firmware tales como sensar, monitorear, almacenar información, etc.

Para ello en la Figura 17 y Figura 18, se representa básicamente cuatro tarjetas a utilizar, así como los correspondientes dispositivos asociados a cada una de ellas. La tarjeta principal es el Raspberry pi, cuya función será procesar las imágenes de engranajes capturadas desde una cámara, además de contener y correr el servidor web del aplicativo, y por último interactuar bidireccionalmente mediante comunicación serial (UATR) con el sistema Físico (Arduino). El Arduino y su placa de conexiones cumplirá la función de recibir o enviar señales a los sensores o actuadores respectivamente, uno de los actuadores del prototipo es el motor para el movimiento de la cinta de transporte, el cual requiere de una tarjeta para controlar la velocidad del motor a partir de señales PWM (Pulse Width Modulation) generados por el microcontrolador Arduino.

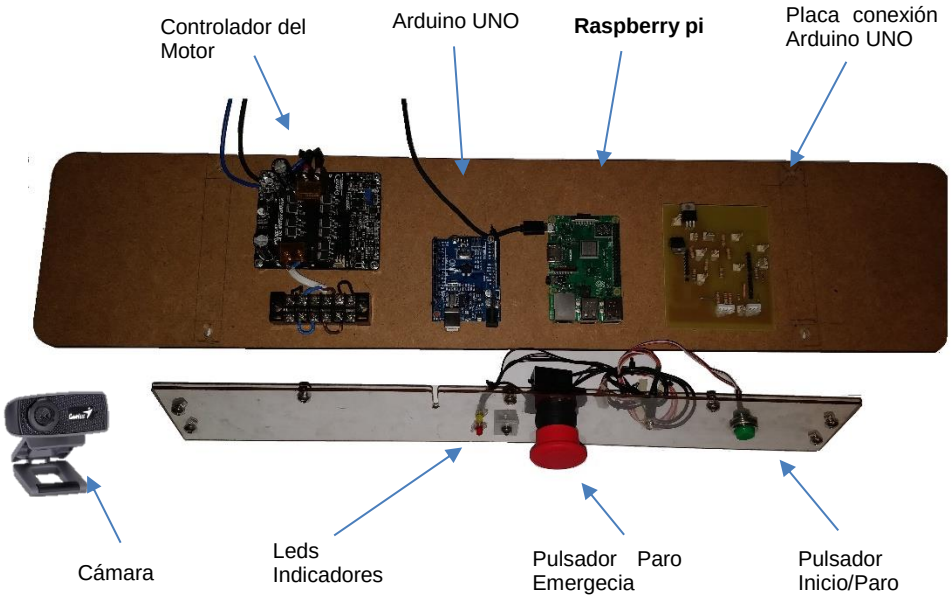


Figura 17. Diseño electrónico.

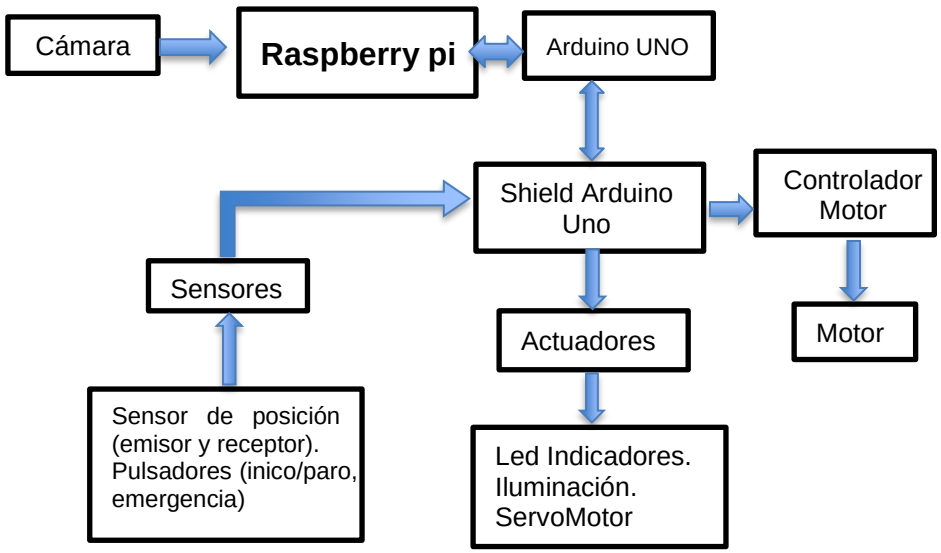


Figura 18. Conexiones del diseño electrónico.

4.3 Hardware

Denominamos hardware a todos los componentes físicos de un sistema, es decir a la parte tangible del equipo, siendo el conjunto de componentes necesarios para conseguir la funcionalidad mínima y especifica del sistema.

En el apartado anterior se mencionaron los diferentes dispositivos que componen el hardware del prototipo SECVIA. Estos dispositivos cuentan con características técnicas necesarias para el funcionamiento idóneo del prototipo. A continuación, se detallarán las especificaciones de los diferentes dispositivos y componente a utilizar.

4.3.1 Actuadores del prototipo.

- **Motorreductor DC.**

Un motorreductor es una unidad compacta y homogénea conformada por un reductor y un motor. Su función principal es reducir la velocidad de giro transmitiendo simultáneamente pares significativos al sistema de cinta de transporte.

Para el funcionamiento del motorreductor es necesario utilizar un driver de control de velocidad del motor, mediante la señal PWM generada por el microcontrolador. En la Figura 19, se muestra al motorreductor con su correspondiente controlador de velocidad a implementar en el prototipo.

El rango de voltaje de trabajo del motorreductor está comprendido entre 12V – 18V y su corriente de consumo mínimo es de 3 Amperios. Además, para el funcionamiento correcto del driver de control de velocidad, se requiere un rango de trabajo máximo rango de trabajo de 30 Amperios y de 5 V – 25V corriente directa.

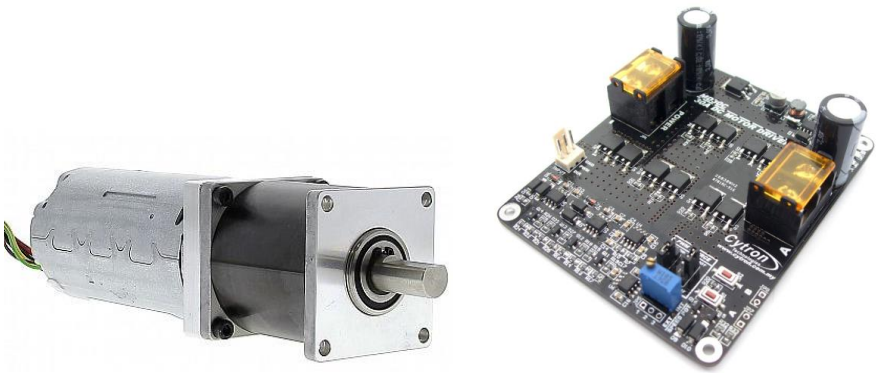


Figura 19. Motorreductor y driver de control de velocidad

- **Servomotor.**

Un servomotor es otro tipo motor especial que facilitá controlar la posición del eje en un momento dado. Diseñado para moverse determinada cantidad de grados y al final mantendiendo fijo su posición establecida. Los servomotores hacen uso de la modulación por ancho de pulsos (PWM) para controlar la dirección o posición de los motores de corriente continua. La mayoría trabaja en la frecuencia de los 50 hertz, así las señales PWM tendrán un periodo de veinte milisegundos. La electrónica dentro del servomotor responderá al ancho de la señal modulada.

El servomotor ayudará a clasificar los engranes buenos y defectuosos, después de recibir información del procesado de la visión de un engrane.

Figura 20. **Servomotor**

- **Diodos led.**

Los diodos led son componentes electrónicos cuya función es permitir el paso de la corriente en un solo sentido, además de emitir luz cuando está polarizado correctamente.

En el prototipo los diodos led a utilizar serán informativos, permitiendo al sistema identificar diferentes estados del funcionamiento como: inicio, paro y paro de emergencia del prototipo. Además de visualizar el estado de la comunicación de datos, procesado del engrane y el sentido de clasificación del engranaje.

Figura 21. **Diodos led Informativos.**

4.3.2 Sensores del prototipo

- **Sensor infrarrojo (Emisor – Receptor).**

El sensor infrarrojo es un dispositivo optoelectrónico capaz de medir la radiación electromagnética infrarroja de los cuerpos. Se basan en la combinación de un emisor y un receptor próximos entre ellos. El emisor es un diodo LED infrarrojo (IRED) y el componente receptor el fototransistor.

Para detectar los engranes que están en movimiento en la cinta de transporte se utilizará la configuración barrera Emisor-Receptor tal como se presenta en la Figura 22, al existir un corte de haz de luz detectado por el receptor, se indicará que un engranaje se encuentra en la cinta de transporte.

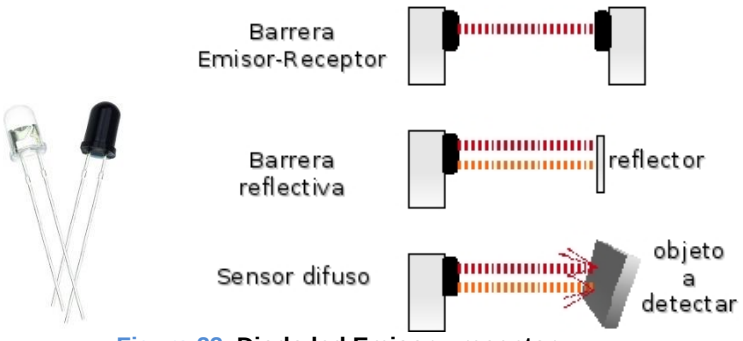


Figura 22. **Diodo led Emisor y receptor.**

- **Sistema Visión.**

Una cámara es un dispositivo tecnológico que tiene como función principal tomar imágenes quietas de situaciones, eventos, etc. A partir de las imágenes capturas mediante el uso de visión por ordenador se logran incluir métodos para adquirir, procesar, analizar y comprender dichas imágenes, revelando información numérica o simbólica.

En el prototipo se utilizará la cámara mostrada en la Figura 23, la misma que se conectará con el Raspberry pi.

Figura 23. Cámara para el sistema de Visión

4.3.3 Elementos de maniobra y control.

Los elementos de maniobra y control son dispositivos que permite realizar operaciones de apertura o cierre de un circuito cuando uno lo requiera. En este trabajo se utilizará un pulsador normalmente abierto para el control del funcionamiento del sistema físico, cuya función será iniciar el sistema o parar el sistema, además de contar con un pulsador normalmente cerrado con enclavamiento para el uso de un sistema de paro de emergencia.

Figura 24. Mandos de control.

4.3.4 Arduino

La plataforma Arduino está basado en una placa sencilla y conformada por entradas y salidas, analógicas y digitales, bajo un entorno de programación Wiring y de desarrollo Processing. En si Arduino cuenta con un microcontrolador Atmel AVR programable capaz de realizar operaciones matemáticas a gran velocidad.

Figura 25. Microcontrolador Arduino.

4.3.5 Raspberry pi.

Es un ordenador de placa reducida u ordenador de placa simple (SBC) de bajo coste, en su placa se encuentra montado un procesador, un chip gráfico y memoria Ram. A diferencias de los ordenadores normales no cuenta con disco duro, para lo cual se utiliza una tarjeta SD. El sistema operativo puede ser instalado varios, gran parte basados en el Kernel de Linux. En este proyecto el sistema operativo a utilizar será Raspbian.

Figura 26. Dispositivo Raspberry pi

4.4 Software.

El software es un conjunto de programas, instrucciones y reglas informáticas que permiten ejecutar distintas tareas en un computador, Raspberry Pi va a ser quien cuente con el conjunto de programas correspondiente para la implementación del prototipo, todo el software e inteligencia se situará sobre ella. Estos corresponderán al sistema operativo, algoritmos para el funcionamiento del prototipo, herramientas de ejecución y librerías de aplicación.

4.4.1 Sistema Operativo Raspbian.

Raspbian es una distribución del sistema operativo GNU/Linux, destinado para la placa computadora Raspberry Pi que se orienta a enseñanzas informáticas. Para la instalación del sistema operativo se deberá contar con una tarjeta SD al menos de 8 GB. Es recomendable para esta aplicación utilizar 32 GB, ya que este dispositivo ejecutará varios programas a la vez. Establecido estos requesitos se debe seguir los siguientes pasos:

1. Descargar **Raspbian** en un ordenador común, desde el sitio web de la organización de Raspberry pi (www.Raspberrypi.org).
2. Descomprimir y mediante algún programa se debe quemar o grabar la imagen en la SD card.
3. Finalizado estos pasos, insertamos en la SD card en el dispositivo Raspberry pi.

Para visualizar el escritorio del Raspberry pi desde el ordenador, es necesario la utilización de un acceso remoto mediante la aplicación RealVNC.

Figura 27. **Escritorio de Raspberry pi utilizando VNC.**

La utilización de un interfaz gráfico dentro de un microprocesador consume recursos innecesarios que se podrían ser útiles para otras funciones. Para este caso la instalación de las herramientas se realizará desde el terminal o consola del sistema.

4.4.2 Instalación de herramientas y librerías.

El siguiente paso es instalar herramientas para la creación del servidor, para esto se utilizará de Node.js como motor del servidor están relacionados con el internet de las cosas, en especial aquellos que tienen gran dependencia de datos en tiempo real. Antes que nada, el dispositivo debe estar conectado a internet vía wifi o cable, para descargar, instalar y actualizar al dispositivo.
Para añadir esta herramienta se debe aplicar en la consola del Raspberry pi los siguientes comandos:

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get install nodejs npm
```

Ahora el dispositivo cuenta con el motor para ejecutar el servidor, un entorno de programación basado en el lenguaje ECMAScript, asíncrono, con I/O de datos en una arquitectura orientada a eventos.

Para lograr integrar el prototipo dentro de un sistema Ciber-físico, el microprocesador Raspberry pi debe contar con las librerías que permitan la facilidad de comunicación y procesamiento de imágenes, que son detalladas en los siguientes apartados.

- **OPEN CV.**

Esta librería permite visualizar y procesar imágenes.

```
sudo apt-get install libgtk2.0-dev pkg config
```

- **Librerías de Boost.**

Librería que ayuda a extender funciones de códigos desarrollados en C++, en este trabajo se utilizará todas las librerías de Boost, esta librería es requerida por la ejecución de socket io.

```
sudo apt-get update
sudo apt-get install libboost-all-dev
```

- **Repositorio RAPIDJSON.**

Rapidjson es un código abierto, que se utiliza como biblioteca de C++ solo para encabezados. RAPIDJSON es un generador y analizador de arreglos Json para el intercambio de datos ligeros. [31] El prototipo debe contar con el repositorio para esto hay que clonarlo y se utiliza el siguiente comando:

```
git clone https://github.com/Tencent/rapidjson.git
```

- **Repositorio WebSocket.**

WebSocket ++ es una biblioteca de C ++ de encabezado único que implementa el Protocolo WebSocket. Permite integrar la funcionalidad de servidor y cliente WebSocket en programas C ++. Utiliza módulos de transporte de red intercambiables, incluido uno basado en buffers de caracteres brutos, uno basado en iostreams de C ++ [32]. De la misma manera se debe clonar el repositorio al prototipo.

```
git clone https://github.com/zaphoyd/websocketpp.git
```

Para que estos repositorios tengan su funcionalidad dentro del sistema se debe compilar e instalar dentro del sistema operativo. En la consola se debe ejecutar los siguientes comandos individualmente

```
Downloads $ cd rapidjson/
Downloads/rapidjson $ cd build
Downloads/rapidjson/build $ make
Downloads/rapidjson/build $ sudo make install


Downloads $ cd websocketpp/
Downloads/websocketpp $ cd build/
Downloads/websocketpp/build $ make
Downloads/websocketpp/build $ sudo make install
```

4.4.3 Desarrollo entorno ClienteSocket (Sistema Físico).

El prototipo físico será considerado como un cliente, la conexión del cliente se desarrollará basado en programación C++, para esto en el directorio del proyecto debe incluirse las

librerías correspondientes a la aplicación detalladas a continuación y se utilizará el siguiente algoritmo tal como se muestra en la Figura 28.

```
#include "sio_client.h"       //Librería de Sockets io
#include "base64.h"           //Libreria para transformar strings a base64
#include <functional>         //Librerías estándar
#include <iostream>           //Librería estándar para operaciones I/O
#include <thread>             //Librerías de Multiprocesamiento
#include <mutex>              //Evita bloqueo de Subprocesos
#include <condition_variable>
#include <string>             //Librerías de cadena de caracteres
#include <fstream>            //Librería para abrir archivos
#include <vector>             //Manejo de vectores
#include <chrono>             //Librería para fecha y año
#include <cstdlib>
#include <stdio.h>           //Librerías estándar
#include <errno.h>           // comporbación de errores
#include "ArduinoJson.h" //Librería para el Manejo de Cadenas Json en C++
#include "Vision.h"      // acceso al archivo de procesamiento de engranajes
```

Para habilitar la comunicación UART del microprocesador Raspberry pi, la librería WiringPi permite acceder a los pines de entrada y salida de la Raspberry, debe estar incluida estas librerías internas del procesador, logrando así que el microprocesador y el microcontrolador estén comunicados. Tal como se muestra a continuación:

```
#ifdef __arm__
      #include <wiringPi.h>
      #include <wiringSerial.h>
#endif
```

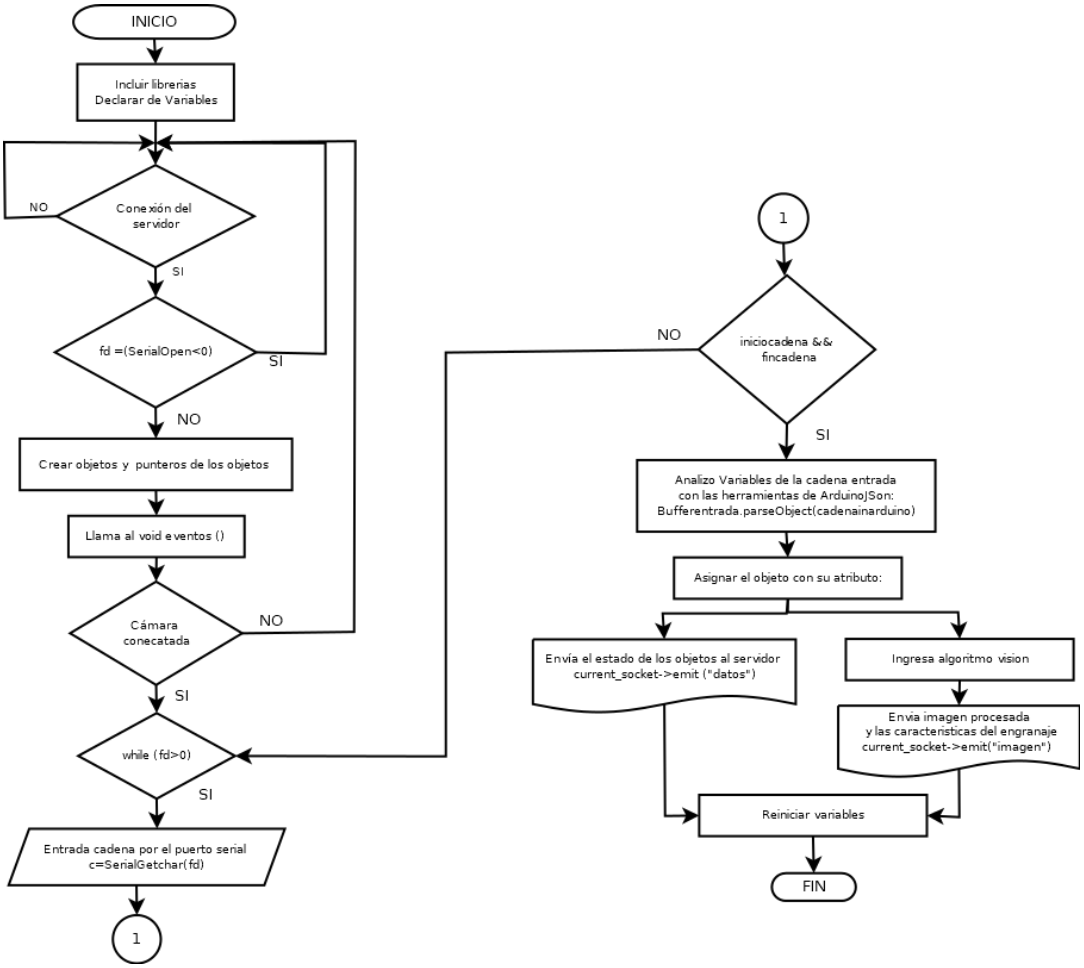


Figura 28. Algoritmo del cliente (Sistema Físico SECVIA) desarrollado en C++

El algoritmo del cliente, en algún momento llamará al contexto eventos de los objetos creados en el servidor, para luego ser enviados mediante comunicación serial desde el microprocesador Raspberry pi hacia el microcontrolador Arduino.

Los objetos que llegan del servidor se encuentran dentro de una matriz, en el contexto eventos, identificará y asignará los diferentes objetos correspondientes a su funcionalidad y para ser enviados al microcontrolador. Los objetos para utilizar se presentan en la Tabla 2

OBJETO	Atributo
dispositivo:	int
modofunc	bool
motor	int
presencia:	bool
estado:	int
imagenprocesada	bool
clasificador	bool
sentidomotor	int
direcclasificacion	int
capturarengrane	bool
umbral	int

Tabla 2. Objetos y atributos recibidos del servidor

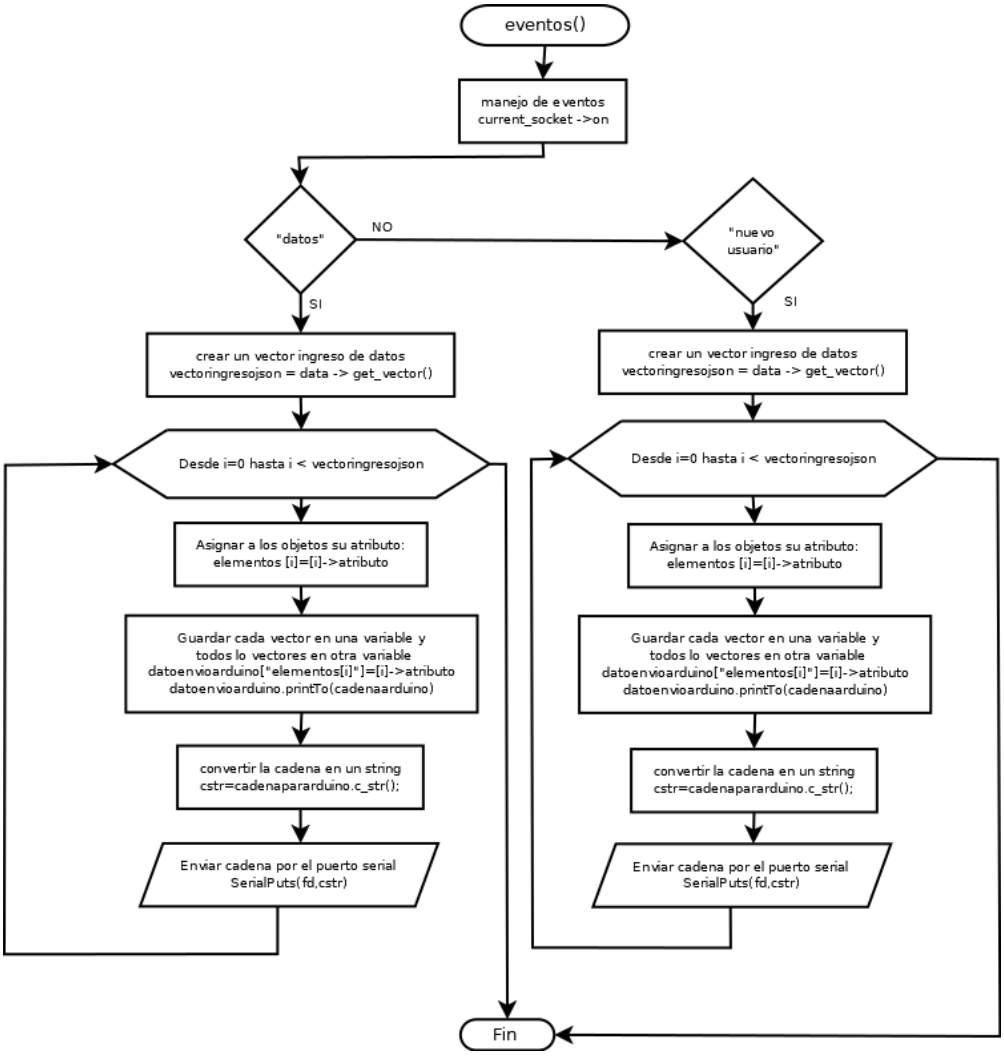


Figura 29. Eventos recibidos del servidor y enviados al Arduino.

4.4.4 Algoritmo de Visión

En este trabajo se integran funciones de visión artificial. El prototipo está enfocado a detectar engranes rectos y verificar si dicho engrane cuenta con los dientes correspondientes, así como datos técnicos del engranaje procesado. Estos datos son obtenidos a partir de ecuaciones de los fabricantes para engranes rectos. En función al estado del engrane, el microprocesador Raspberry pi enviará la información de clasificación al microcontrolador Arduino y consiguientemente al actuador.

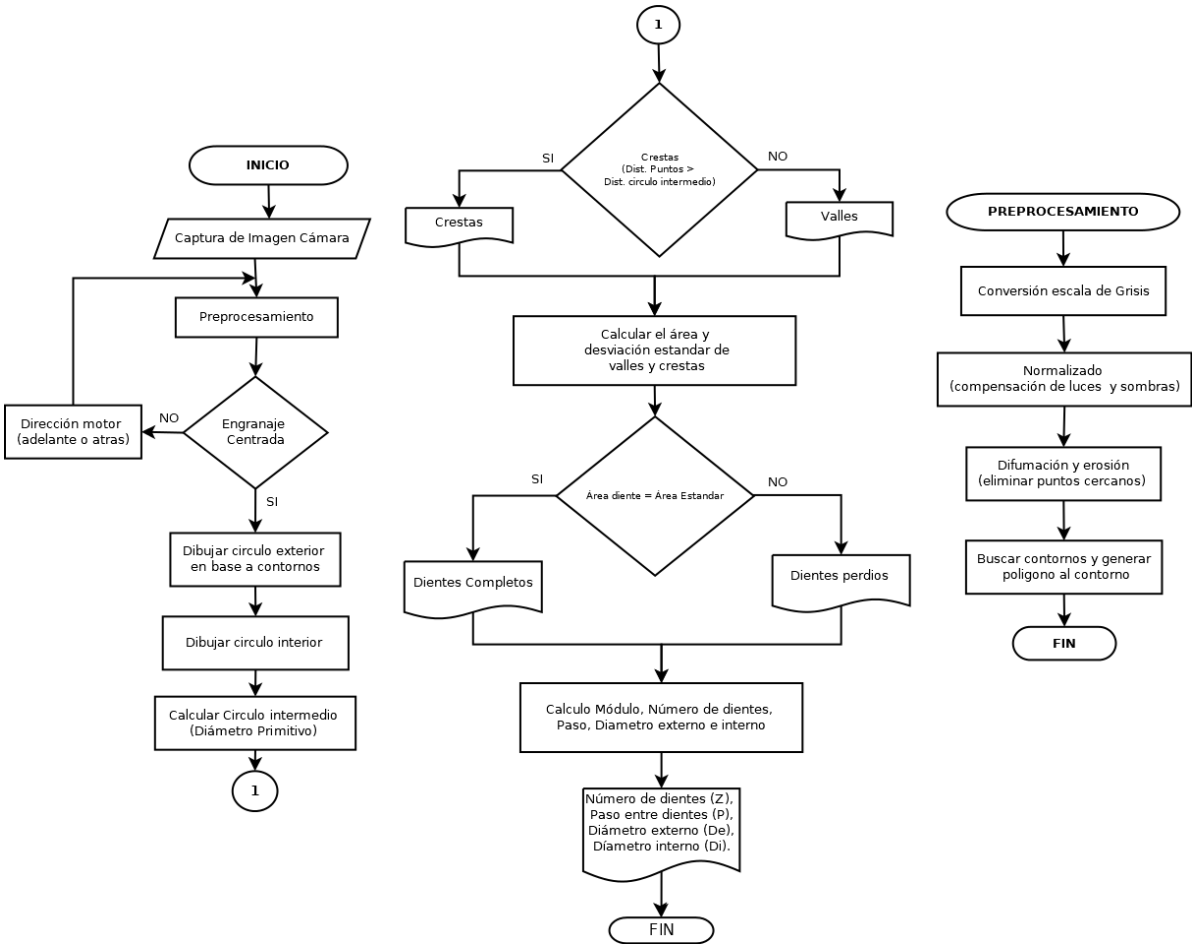


Figura 30. Algoritmo de detección fallas engranajes

4.4.5 Desarrollo entorno Servidor y Sitio Web.

- **Desarrollo sitio Web.**

Un sitio Web está compuesto por diferentes enlaces a diferentes documentos, cada documento extensión .HTML cuenta con las funciones necesarias para el interfaz con el usuario. Estos documentos deben estar dentro de un directorio del proyecto. Para este prototipo el directorio estará dentro de **ServidorWeb.**

Los encabezados, navegadores y pies de página de enlace dentro de los documentos en la mayor parte son los mismos, debido a que es la información y enlace del sitio web y la ventana en la que el usuario está ubicado. El cuerpo de cada documento es el lugar donde contendrá el correspondiente contenido propios del servidor y funciones que se interactuará

con los sistemas físicos. En la Tabla 3, se muestra los diferentes contenidos y funciones que contendrá cada documento html del sitio web.
En la Figura 31 se muestra la estructura de los diferentes documentos con extensión html, para para la aplicación.

Documento	**Nombre de la ventana**	**Cuerpo del documento <body>**
index.html	INGRESO DE USUARIOS	Espacio para el ingreso del usuario y la contraseña
inicio.html	INICIO	Información del prototipo y de los CPS
controlymoni.html	CONTROL Y MONITORIZACIÓN	Mandos de control automático y manual, visualización resultados de la imagen procesada.
historial.html	HISTORIAL	Información de los engranajes procesados, datos técnicos, Errores de los engranajes y fecha de procesado.
estadisticas.html	ESTADISTICAS	Información porcentual del número de engranajes procesados, además de indicar el número de engranes validados como buenos defectuosos.
doc.html	DOCUMENTACION	Información acerca de los CPS y del prototipo.

Tabla 3. Contenido y funciones del cuerpo de los documentos HTML.

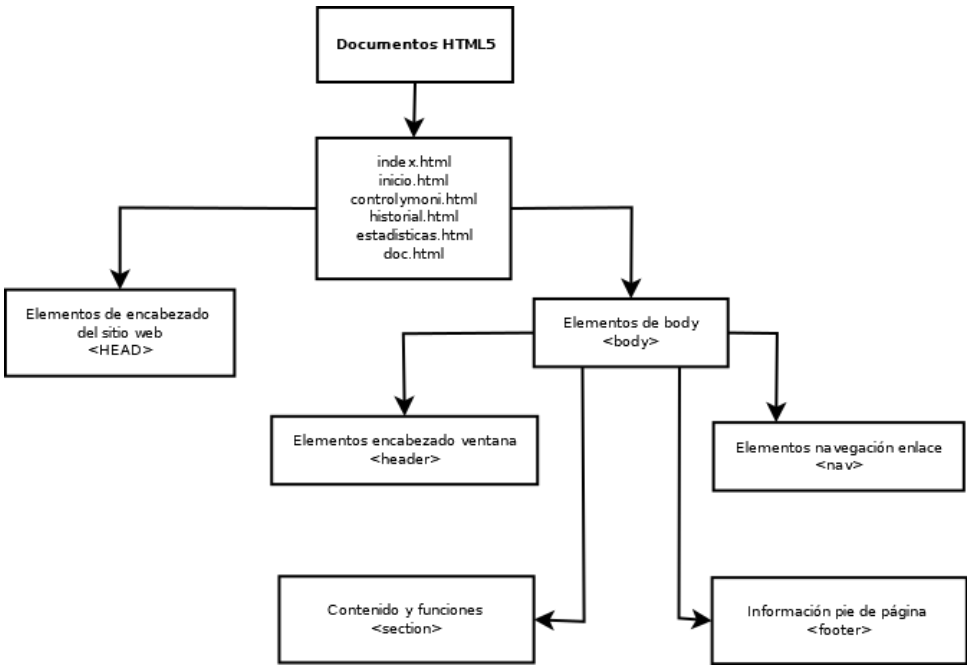


Figura 31. Estructura de Sitio Web del aplicativo del prototipo.

- ## Estilos dinámicos con JavaScript.

El entorno de programación de JavaScript permitirá que los elementos de los documentos html sean dinámicos para el usuario, además de la conexión de los diferentes eventos hacia el cliente mediante los arreglos JSON. Hay que recordar que el aplicativo servidor – cliente enviará y receptará arreglos JSON, como se muestra en la siguiente sintaxis:

```
var dispositivo = {objeto_1: 1, objeto_2: false };
```

En la Figura 32, se muestra el contenido de los ficheros main.js correspondiente al estilo dinámico del aplicativo web, además del fichero app.js del servidor web y que permite la conexión con el cliente, estos ficheros estan bajo el entorno de programación JavaScript.

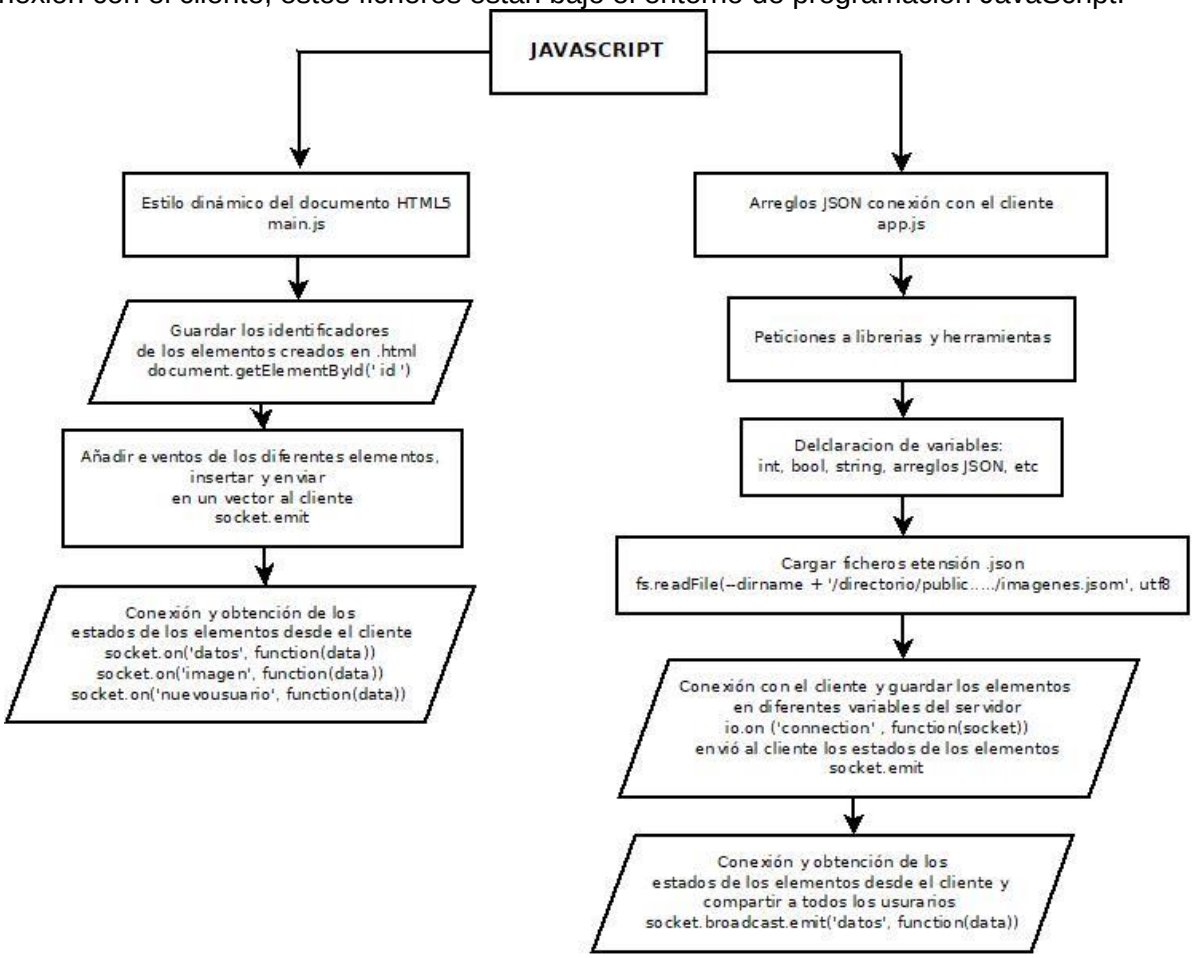


Figura 32. Estilos Dinámicos y conexión cliente-servidor con JAVASCRIPT

Establecido las estructuras en los entornos de programación, estas estructuras interpretaran unas a otras al mismo tiempo, logrando tener un sitio web desarrollado y completo. En la Figura 33, se representa los ficheros que son enlazados con el aplicativo web y el servidor web.

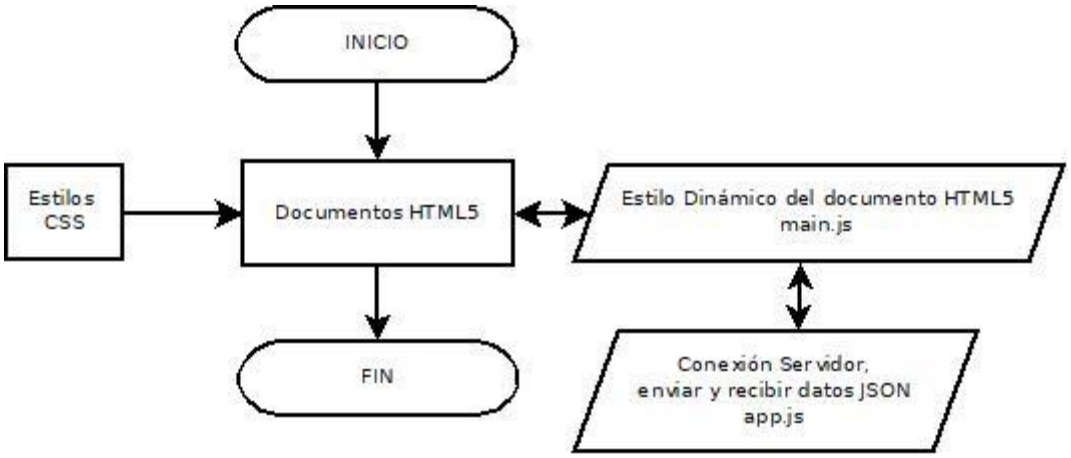


Figura 33. Sitio web con sus herramientas y motores de funcionamiento.

- **Desarrollo servidor**

El prototipo estará montado con la herramienta Node.js, que permite escuchar peticiones HTTP por un puerto (ejemplo 3000), y responder a las peticiones de objetos orientados a Notación (JSON). Para esto la necesidad de la creación de un proyecto de aplicación HTML/JS con Nodejs, además con sus herramientas express y package.json. A continuación, se detalla los diferentes documentos con su respectivo entorno de programación:
El documento package.json, guardará información del Servidor, así como la habilitación de las diferentes dependencias a utilizar en el servidor.

```
{
  "name": "ServidorWeb",
  "version": "0.0.1",
  "private": true,
  "author": "Fernando_Jacome",
  "scripts": {  "start": "node bin/www" },
  "dependencies": {
    "body-parser": "~1.0.0",
    "cookie-parser": "~1.0.1",
    "debug": "~0.7.4",
    "express": "~4.0.0",
    "jade": "~1.3.0",
    "less-middleware": "0.1.15",
    "socket.io": "^2.1.1",
    "static-favicon": "~1.0.0"
  }
}
```

Ahora bien, la aplicación ya cuenta habilitado las funciones y herramientas, luego se debe realizar peticiones http, a la aplicación, y a la vez configurar el puerto de comunicación. Y por último se crea el servidor de la aplicación.

```
var app = require('../app');
var debug = require('debug')( ServidorWeb:server');
var http = require('http');
var port = (process.env.PORT || '3000');
app.set('port', port);

var server = http.createServer(app);
server.listen(port);
```

5. RESULTADOS

En este capítulo describe los resultados conseguidos con el prototipo SECVIA, desde el amabito funcional al académico.

5.1 Estación de trabajo académica.

En la Figura 34, muestra el sistema físico del prototipo académico SECVIA implementado, con sus correspondientes dispositivos hardware y software que permitirá crear un entorno CPS.

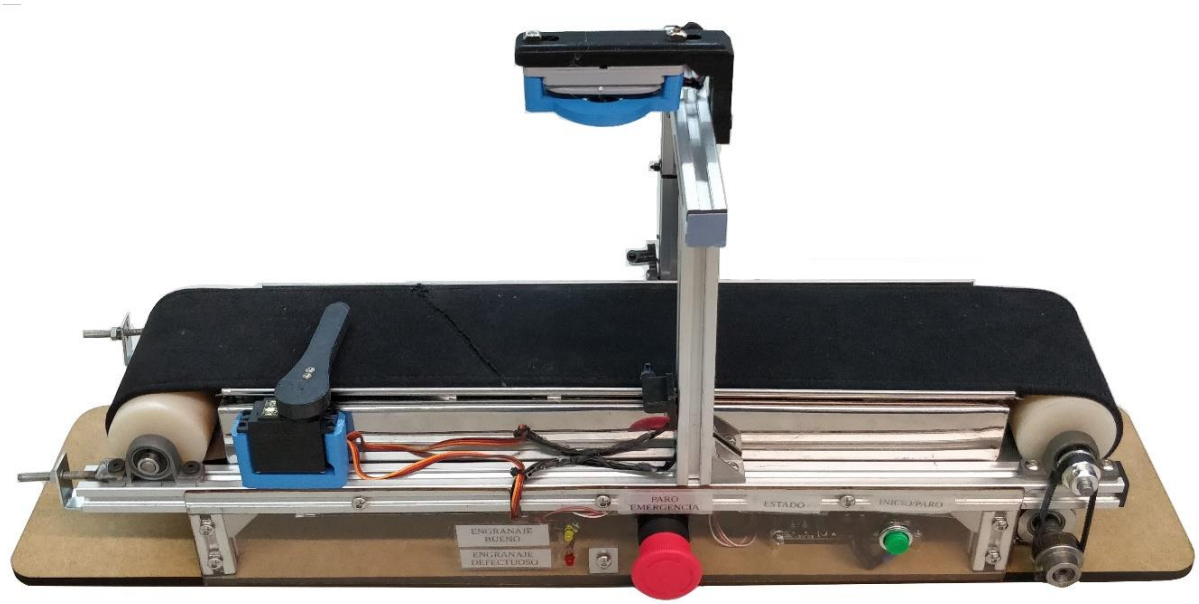


Figura 34. Prototipo académico SECVIA

5.2 Procesamiento de un engrane mediante visión.

En el algoritmo de visión del prototipo para el procesamiento de la imagen capturada por una cámara, hay que considerar los kernel o filtros necesarios, con el objetivo de visualizar los dientes de un engrane si ningún error. También debe considerar el ambiente de trabajo en el cual se está aplicando el procedimiento del procesado, es decir la iluminación debe ser adecuada para un buen procesamiento del engranaje.

Establecido la iluminación adecuada y aplicado los filtros necesarios para convertir a una imagen tipo binaria, en la Figura 35 se muestra un error común que es producido por el brillo excesivo, y que es reflejado por el material en el cual los objetos(engranajes) son transportados.

Dentro de las primeras pruebas de procesamiento y clasificación, se determinó que el material que permite el trasporte de los engranajes presentaba una retroalimentación de la iluminación, lo que provocaba que el algoritmo de visión considere parte del engranaje.

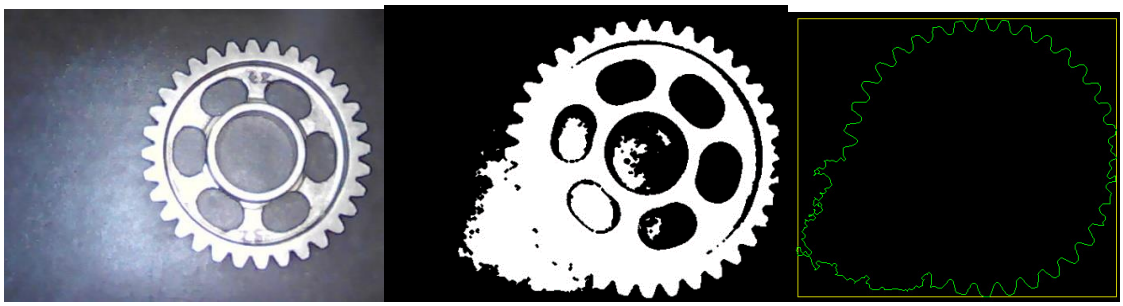
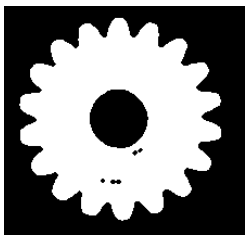
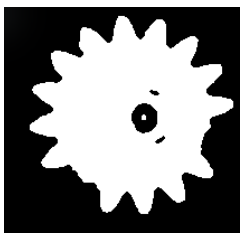
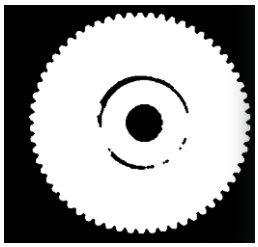
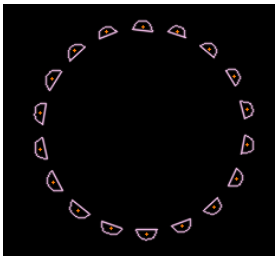
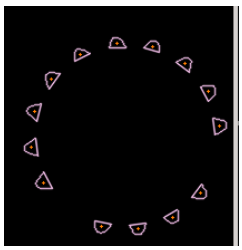
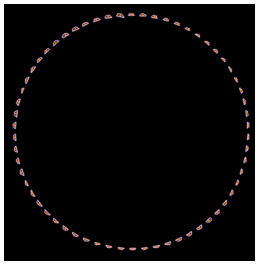


Figura 35. Errores provocados por reflejos de la banda de transporte.

Considerado este último punto, para lograr solucionar el material que cumple con la función de transportar los engranajes, se vió la necesidad de implementar un recubrimiento de un material antirreflejante de la iluminación.

Tomado en cuenta los puntos negativos para el procesamiento de un engranaje, en la Tabla 4 se muestra los resultados de los diferentes ensayos realizados a diferentes engranajes.

ENSAYO #1	ENSAYO #2	ENSAYO #3
IMAGEN CAPTURADA		
		
IMAGEN PROCESADA Y BINARIA		
		
DETERMINACIÓN DE DIENTES		
		

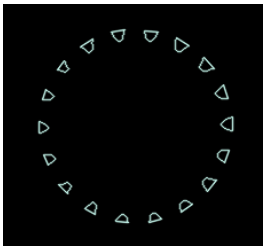
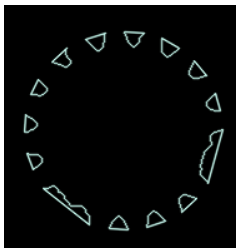
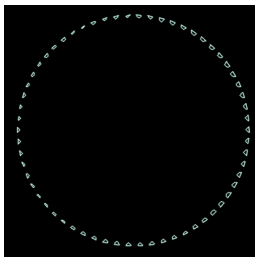
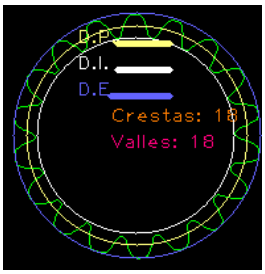
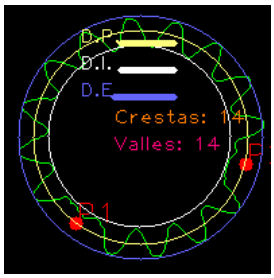
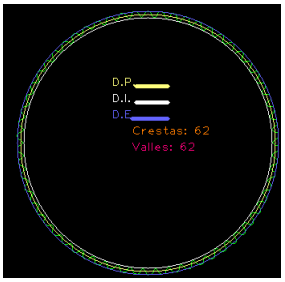
DETERMINACIÓN DE VALLES		
		
RESULTADO DEL ENGRANAJE		
		
*****Información del engrane***** Hay 0 dientes perdidos * Diámetro exterior (De): 39.1782 mm * Diámetro interior (Di): 31.5568 mm * Modulo(M): 2 * Paso(P): 6.28319 * Número de dientes(Z): 18	*****Información del engrane***** Hay 2 dientes perdidos * Diámetro exterior (De): 38.0673 mm * Diámetro interior (Di): 28.6776 mm * Modulo(M): 2 * Paso(P): 6.28319 * Número de dientes(Z): 16 ********** P1 No hay diente P2 No hay diente	*****Información del engrane**** *Hay 0 dientes perdidos *Diámetro exterior (De): 68.3076 mm *Diámetro interior (Di): 64.7308 mm * Modulo(M): 1 * Paso(P): 3.14159 * Número de dientes(Z): 62

Tabla 4. Resultados de visión artificial en engranajes rectos.

5.3 Aplicativo del cliente desde un sitio Web.

Mediante la utilización de un sitio web con un dominio conocido por el cliente se puede enlazar al servidor del prototipo, y así lograr visualizar los diferentes entornos de control y monitorización, un historial de los engranajes procesados y las estadísticas de todos los datos procesados.

En la Figura 36 se muestra estilo del sitio web, así como la ventana control y monitorización del engranaje que ha sido procesado por el sistema ciber físico. Sitio web que podría ser monitorizada desde cualquier punto del internet por medio de un dominio conocido, debido a que el prototipo tiene fines educativos y de demostración, el domino será remplazado con una dirección ip con su respectivo puerto de comunicación, que es generado por el servidor que está conectado a la red interna de la institución.

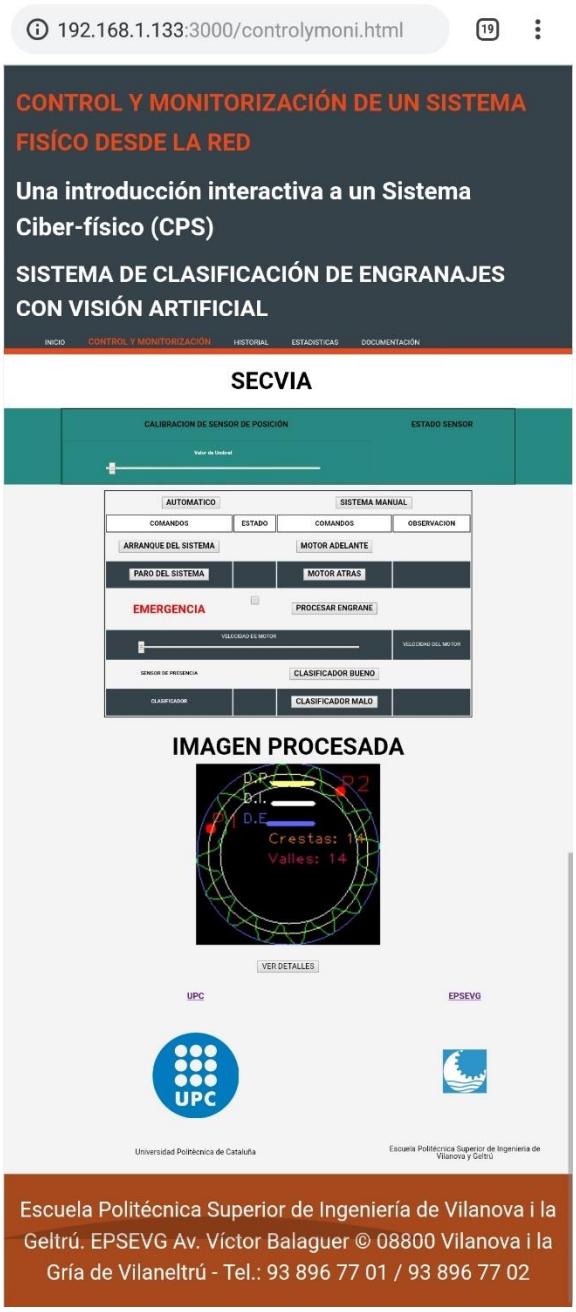


Figura 36. Estilo del Sitio web del aplicativo

Debe tenerse en cuenta una consideración con el servidor web, si el cliente envía información inadecuada, haciendo referencia a datos técnico de las imágenes procesadas, el subsitio web historial y estadísticas reflejarán errores, provocando que el servidor se suspenda hasta que sea verificado los errores que provocaron esta suspensión.

6. GUÍA PRÁCTICA DE LABORATORIO

6.1 Introducción.

La principal finalidad de este trabajo es el desarrollo de un entorno Ciber-físico que permita la realización de prácticas de laboratorio para alumnos del Máster Universitario en Ingeniería de Sistemas Automáticos y Electrónica Industrial de la EPSEVG de la Universidad Politécnica de Cataluña y en el resto de las instituciones que imparten enseñanzas sobre los sistemas ciber-físicos.

Para esto se contará con un prototipo de fines académicos, en el cual se podrá lograr interpretar un sistema físico y un sistema enfocado a un Servidor-Cliente, estos estarán conformados por aplicaciones de importancia para el desarrollo del entorno ciber-físico. Aplicaciones que están destinadas tanto para realizar eventos o alguna manipulación en el servidor Web o desde el sistema Físico.

Por lo tanto, al contar con una página diseñada para la aplicación se debe diseñar e implementar dentro de una ventana, específicamente en la ventana Control y Monitorización de la Página Web, conformado por comandos de controles de forma manual de los distintos actuadores del sistema con la ayuda de los entornos de programación mencionados en este trabajo, con el objetivo de obtener información del proceso del sistema físico en tiempo real.

Se debe tomar en cuenta para el diseño de la aplicación, la creación de estructuras de objetos en función a las librerías de socket.io y ArduinoJson, las mismas que ayudarán a comunicarse desde la plataforma del Servidor Web, la plataforma Raspberry y la plataforma Arduino de forma simultánea, logrando obtener como punto de partida una aplicación del internet de las cosas (IoT).

Como este trabajo está enfocado a un sistema Ciber-físico, se deberá diseñar un algoritmo de visión artificial, y aplicado al control de calidad. Dando un paso hacia adelante de las IoT e integrando sistemas computacionales.

6.2 Objetivos:

A partir de un sistema monitorizado y controlado automático se debe:

- Controlar vía remota actuadores del sistema de forma manual, de esta manera el usuario adquirirá controlar individualmente a los distintos actuadores del prototipo desde algún sitio de intranet o internet.
- Implementar comandos individuales para el control de los actuadores.
- Crear y enviar de cadena de objetos basándose en ArduinoJson.
- Interpretar las aplicaciones del Servidor (Raspberry pi) y del sistema físico (Arduino)
- Leer y enviar cadena de objetos en la plataforma Arduino.

6.3 Estación de un CPS y aplicativo web.

6.3.1 Prototipo académico SECVIA.

El prototipo SECVIA presentado en la Figura 37, es un sistema académico de aprendizaje y que está orientado al servicio de control de calidad de engranajes mediante visión. Este trabajo consiste en dos partes importantes para el aprendizaje de un CPS; la manipulación de un sistema físico a escala y la implementación de modificaciones al software y hardware para el control y procesamiento de engranes rectos.

Uno de los objetivos es comprender el comportamiento de un CPS, cuando el usuario interactúa con el sistema físico desde cualquier punto de la red. El trabajo se centrará dentro de una red de una organización (intranet). Para el funcionamiento del prototipo se debe considerar las siguientes características técnicas presentadas en la Tabla 5.

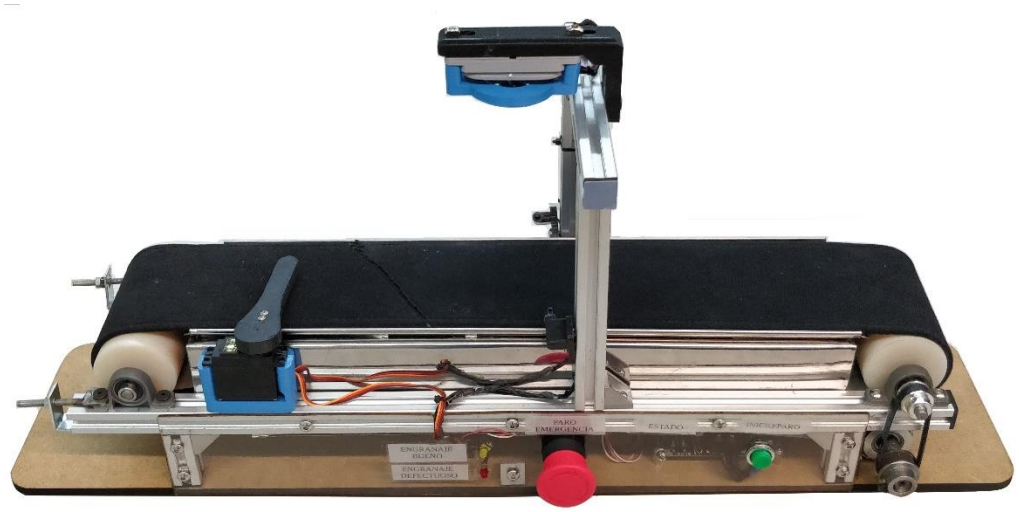


Figura 37. Estación de clasificación SECVIA

Ítem	Sistema	Alimentación de entrada
1	Control del prototipo SECVIA	6V - 12V DC, min. corriente de 5 Amperios
2	Microprocesador Rasperry pi	5V DC, min corriente de 2,5 Amperios

Tabla 5. Características de Funcionamiento del prototipo SECVIA

Las fuentes de alimentación de cada uno de los sistemas deben estar alimentadas individualmente, para evitar que el sistema de control generé variaciones de corriente y voltaje que puede afectar al microprocesador.

6.3.2 Entorno cliente-servidor.

Para el desarrollo de un cliente servidor se utiliza un microprocesador. En este proyecto se utilizará un Raspberry pi, quien va a ser quien ejecute el servidor web, además de interactuar con los diferentes dispositivos, debido a que contará con los diferentes softwares y programas instalados para el desarrollo de un sistema CPS.

Para el desarrollo de este prototipo SECVIA se debe contar con conocimientos básicos de entornos de programación HTML, JavaScript, C++, y C para Arduino. Establecido estos requerimientos básicos, la necesidad de diferenciar directorios que se enfocan al servidor y a la conexión del cliente, a continuación, se detalla de manera rápida como el usuario ingresará al ordenador del Raspberry pi desde la consola de otro ordenador.

1. Conocer la dirección ip del ordenador del usuario, si el usuario desconoce su dirección ip, en la consola del ordenador del usuario se aplica los siguientes comandos, recomendable el uso de un ordenador con Sistema operativo en Linux, ya que el sistema operativo del Raspberry pi está basado en el kernel de Linux.

```
ifconfig
```

En la Figura 38 se muestra el resultado de la dirección ip del usuario que está utilizando en la red.

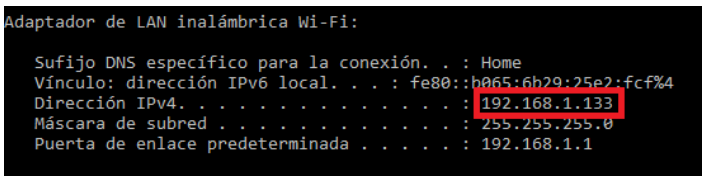


Figura 38. Identificación de la dirección ip del usuario.

2. El siguiente paso es realizar un mapeo de las direcciones IP de los diferentes dispositivos que están conectados en la red mediante el comando.

```
nmap -sn 192.168.1.133/24
```

3. Identificado la dirección ip del microprocesador Raspberry pi, al aplicar el primer comando de la Tabla 6, se puede ingresar al sistema operativo del Raspberry pi. Mediante la consola o bien desde un explorador de archivos.

- Utilización de la consola. Al utilizar este método, el estudiante debe estar consciente que va a trabajar en adelante desde el terminal. Para eso debe conocer las diferentes funciones de uso en una consola. En la Tabla 6 se muestra las funciones básicas que se utilizará para el desarrollo del proyecto.

Comandos	Función
`sudo ssh pi@192.168.1.133`	Ingresa al sistema operativo del Raspberry pi luego de solicitar la contraseña correspondiente para el ingreso. Clave: raspberry
`cd Documentos`	Ingresa al directorio Documentos
`cd Documentos/UPC_TFM_Fernando_Jacome/ServidorWeb`	Ingresa al directorio del proyecto.
`ls`	Despliega todos los archivos del directorio.
`nano controlymoni.html`	Ingresa al documento mencionado, lugar donde se encuentra los diferentes entornos de programación que podrán ser modificados.

Tabla 6. Comandos útiles para consola de Raspberry pi

- Utilización de un explorador de archivos, mediante la utilización de la siguiente función fish://192.168.x.xx, que facilitará el ingreso y una fácil visualización del contenido de los diferentes ficheros que se encuentra en el sistema operativo Raspberry pi.

Hay que tomar en cuenta, que para acceder al sistema operativo y consiguiente a sus directorios del Raspberry pi, el dispositivo solicitará su nombre y su contraseña. Estos están definidos por defecto de fábrica User: pi; Password: raspberry.
En la Figura 39 se muestra los diferentes ficheros después de ingresar al ordenador de Raspberry pi. Para el desarrollo de las prácticas se utilizará el directorio UPC_TFM_Fernando_Jacome, que está contenida en Documentos del Raspberry pi.

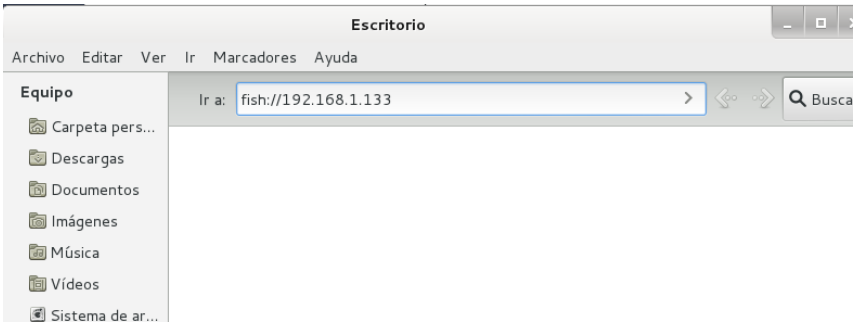


Figura 39. **Escritorio de Raspberry pi.**

4. En directorio mencionado de la aplicación se despliega los directorios tanto para el servidor y el cliente. En la Figura 40 se muestra los diferentes ficheros y directorios de la aplicación. Los ficheros de la carpeta de ClienteSocket se encuentra los diferentes algoritmos para visión y la comunicación con el sistema físico. En la carpeta ServidorWeb, contiene los diferentes ficheros que permitirán enviar los diferentes eventos al sistema físico y bien visualizar dentro de un subsitio web estos cambios de los eventos.
   Y, por último, la carpeta TFM_ARDUINO contiene el algoritmo del sistema físico que interactuará con los diferentes sensores y actuadores
   De aquí en adelante nos enfocaremos en la carpeta del ServidorWeb para realizar las diferentes prácticas.

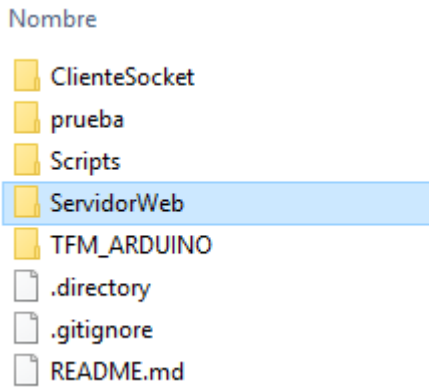


Figura 40. **Ficheros del proyecto SECVIA**

5. Al desplegar el directorio mencionado en el anterior apartado, el directorio cuenta con ficheros de las herramientas y librerías para que el servidor funcione correctamente, los estilos de la aplicación que se encuentran dentro del directorio de la carpeta **public**. En la Figura 41 se muestra los diferentes ficheros para crear el estilo, estilos dinámicos y contenidos del sitio web.

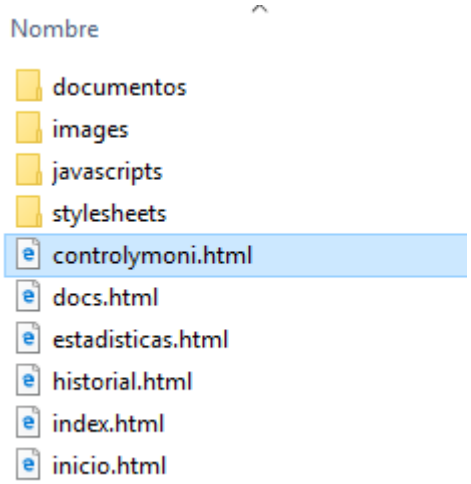


Figura 41. **Ficheros de estilos y contenidos del sitio web de la aplicación.**

Establecido e identificado los diferentes ficheros de la aplicación, ahora bien, queda en enfocarse en diseñar e insertar el contenido y comandos deseados por usuario. Esta guía práctica contará con dos laboratorios para lograr comprender el funcionamiento de un CPS.

6.4 Guía de laboratorio #1. Controlar un sistema de forma Manual.

6.4.1 Implementación de comandos en la página web.

A. Creación de comandos en el entorno de programación HTML.

El diseño actual del subsitio web control y monitorización cuenta con comandos para trabajar de forma automática tal como se muestra en la Figura 42, sin interferir en el estado y posición que se encuentre cada actuador, como objetivo principal dentro de esta parte de la guía práctica, es aprender a integrar comandos individuales de cada actuador asociado al sistema.

Figura 42. **Diseño de comandos para un sistema automático**

Ahora bien, para crear una tabla dentro de un fichero HTML, en primer lugar, hay que Interpretar el número de columnas y filas que cuenta la tabla insertada del sistema automático, para este caso 3 filas por 7 columnas. Tomando en cuenta que la estructura para crear tablas en documentos con extensión html, a continuación, se observa la sintaxis de este entorno de programación:

Columna 1	Column 2	Column 3
A	B	C
D	E	F

Tabla 7. Estructura de una Tabla en un Sitio Web.

```
<TABLE BORDER>
        <TR>
                <TD>Column 1</TD> <TD>Column 2</TD> <TD> Column 3</TD>
        </TR>
        <TR>
                <TD>A</TD> <TD>B</TD> <TD>C</TD>
        </TR>
        <TR>
                <TD>D</TD> <TD>E</TD> <TD>F</TD>
        </TR>
</TABLE>
```

Se puede observar que para construir en HTML una fila se utiliza la terminología `<TR></TR>`, consiguientemente dentro de la fila se inserta las columnas usado la terminología `<TD></TD>`. Cabe notar que dentro de cada terminología de columna se utilizará el campo para crear botones o contenido texto.
Para este ejercicio se creará una columna en cada fila y los diferentes comandos a utilizar son:

Tipo	Nombre del Botón	Identificador del botón (id)	Identificador del texto (id)
Button	SISTEMA MANUAL	id="manual"	
Button	MOTOR ADELANTE	id="motoradelante"	Id="motoradelantelabel
Button	MOTOR ATRÁS	id="motoratras"	id="motoratraslabel"
Button	CAPTURAR Y PROCESAR ENGRANE	id="capturarengrane"	id="textcapturarengrane"
Button	CLASIFICADOR BUENO	id="clasificacionbueno"	id="engranebueno"
Button	CLASIFICADOR MALO	id="clasificacionmalo"	id="engranemalo"

Tabla 8. Comandos e identificadores para un sistema manual

Los nombres e identificadores se pueden denominar de la manera como el usuario lo desee, para este laboratorio se recomienda utilizar dicha nomenclatura presentado en la Tabla 8, con el fin que no exista confusión con los distintos entornos de programación.

Para la insertar los comandos tipo Button en la tabla construida en el fichero html, estos comandos deberán estar ubicados en la columna número tres, desplazando la columna de OBSERVACIÓN a la columna número cuatro. En la columna cuatro se implementará el espacio para los diferentes textos informativos y para esto se utiliza los identificadores de texto. Dicho espacio contendrá textos informativos tanto para el sistema automático como para el sistema manual.

Una vez insertado los comandos mencionados se podrá obtener una interfaz de control y monitorización dentro del subsitio web como se muestra a continuación en la Figura 43**,** **c**oncluyendo así con el entorno de programación en HTML.

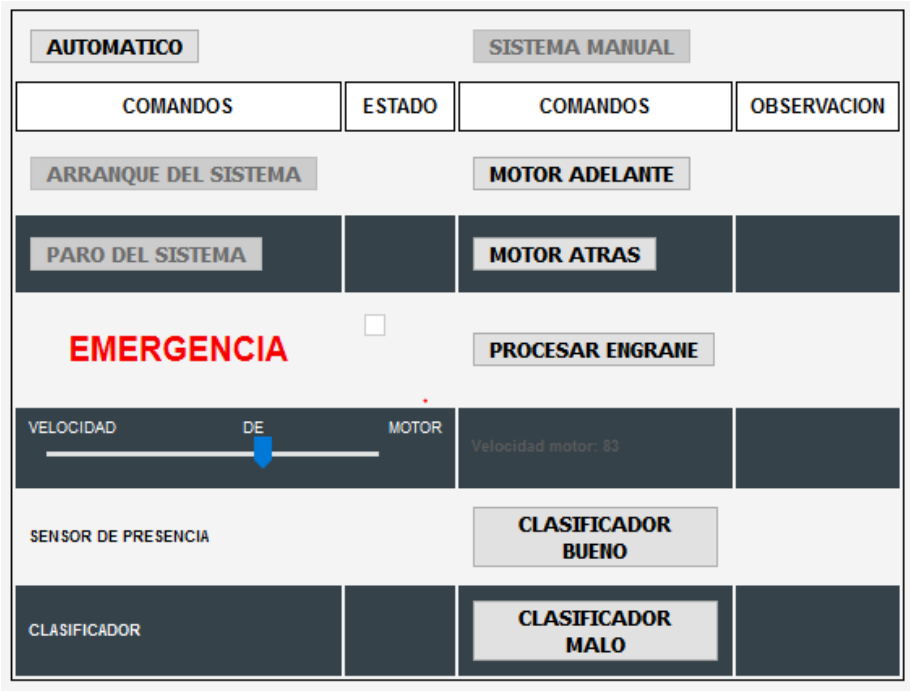


Figura 43. **Diseño del control de un sistema de forma automática y manual**

B. Crear entorno dinámico en JavaScript.

Para crear un sitio web dinámico a los diferentes comandos y contenidos del documento HTML. Se debe recordar los distintos identificadores mencionados en el entorno de programación HTML con el fin de ligarlos con el entorno JavaScript. Estos identificadores llamarán al comando ligado y actuarán dinámicamente en el sitio web y a la vez se creará objetos para enviará mediante un buffer al sistema físico.

Esto se logrará con dos ficheros en el entorno JavaScript **app.js** y **main.js.** Ficheros correspondientes al servidor y al cliente respectivamente.

- **Crear objetos en el servidor bajo la librería socket.io.**

Dentro del fichero **app.js** el primer paso es crear variables para leer y guardar su valor correspondiente. Variables que pertenecerán al servidor de la librería socket.io, considerando el valor a tomar cada variable (int, bool), de la forma presentada:

```
var idservidor_1=0;
var idservidor_2=false;
```

EL fichero cuenta con un arreglo de objetos bajo las librerías Json, este arreglo es de tipo cadena (String) denominada **dispositivo,** esta variable es la encargada de guardar la cadena de objetos que se crea automáticamente después de realizar algún evento dentro la sitio web hacia el cliente y consecuente enviar al sistema físico. En la variable deberá contar con los diferentes objetos incluido su valor, valor que puede tomar como int, bool, string, array.

```
var dispositivo = {objeto_1: 1, objeto_2: false };
```

Para desarrollar la práctica como recomendación se debe utilizar los siguientes objetos, y que serán interpretados en la plataforma Arduino, tal como se presenta en la Tabla 9.

OBJETO	TIPO
dispositivo:	int
modofunc	bool
motor	int
presencia:	bool
estado:	int
imagenprocesada	bool
clasificador	bool
sentidomotor	int
direcclasificacion	int
capturarengrane	bool

Tabla 9. Creación de objetos

Teniendo en cuenta estas variables, mediante de la función **io.on** permitirá establecer la conexión y enviar las variables del servidor que son capturadas desde el arreglo Json con su objeto, como se indica en la sintaxis.

```
io.on('connection', function (socket) {
   if (dispositivo.objeto_1 != null) {
      dispositivo.objeto_1 = idservidor_1
                          }
  if (dispositivo.objeto_2 != null) {
      dispositivo.objeto_2 = idservidor_2;
                          }
```

Para el envío de estos objetos en la función **io.on**, luego de haber guardado las variables mediante el método **push(),** que permite agregar uno o más elementos a un vector (`arreglodispositivo`), y al final emitiendo todo este vector a la conexión del servidor.

```
arreglodispositivo.push(dispositivo);
                dispositivo = {};
socket.emit("nombre_funcion", arreglodispositivo);
                arreglodispositivo = [];
```

Para obtener lectura de algún evento del cliente desde el sistema físico, el procedimiento es leer esa cadena de caracteres que se obtiene a partir de la generación de la estructura por ArduinoJson, la herramienta **socket.on** contendrá una función de datos, estos datos deben ser asociados con los diferentes objetos correspondientes y enviados a todos los clientes con ayuda de las herramientas de socket.on

```
socket.on("datos", function (data) {
       var sizedatos = data.length;
        for (i = 0; i < sizedatos; i++) {
           if (idservidor_1 != null) {
          idservidor_1 = data[i].objeto_1;
                          }
           if (idservidor_2 != null) {
          idservidor_2 = data[i].objeto_1;
                          }
                  }
   socket.broadcast.emit('datos', data);
   });
```

- **Añadir eventos dinámicos de la ventana de la página Web**

Ahora bien, al sitio web se realizará un entorno dinámico para capturar y enviar las diversas respuestas enviadas desde el servidor. Esto se trabajará en el fichero **main.js,** donde se debe crear nuevas variables con el objetivo de obtener y guardar la información del id de los diferentes comandos y textos del fichero .html, mediante la siguiente sintaxis al utilizar en JavaScript es:

```
var id_1 = document.getElementById(id1);
var id_2 = document.getElementById(id2);
```

A la vez se debe considerar una variable tipo vector (Array), variable que guardará los diferentes objetos, y luego enviar a todos los elementos con su respectivo valor al servidor de este proyecto.

```
var arreglodispositivo = [];
```

Para leer los eventos de cada elemento se utiliza la herramienta `elemento.addEventListener()`, herramienta que ayudan a obtener los eventos desde el sitio web. En la Tabla 10, se presenta las diferentes actividades que se puede realizar después de la lectura de un evento.

Herramientas	**Tipo**	**Acción.**
dispositivo.objeto_1	int, bool,array	Valor correspondiente para enviar hacia sistema físico
id_2.checked	bool	Habilitación / deshabilitación permanente un elemento
id_3.disabled	bool	Habilitación / deshabilitación elemento
id_4.style.backgroundColor	Formato .hex	Color de Fondo de los elementos
id_5.style.display	"block" or "none"	Muestra texto de los elementos

Tabla 10. Opciones de la Herramienta addEventListener

El objeto_1 llama al mismo objeto y su valor correspondiente que está contenida en la variable **dispositivo** (Arreglos Json) del fichero app.js, el tipo de valor que puede contener es int, bool, array, etc.

Establecidos los puntos importantes del evento de un elemento la sintaxis a utilizar luego que dicho elemento ha realizado un pulso (“on click”) en el botón de la página web, y al final enviar mediante un vector los objetos hacia el sistema físico.

```
id_1.addEventListener("click", function () {
                var dispositivo = {};
            dispositivo.dispositivo = 1;
            dispositivo.objeto_1 = true;
              dispositivo.objeto_2 = 1;
                 id_1.disabled = true;
            id_2.style.display = "none";
        id_3.style.backgroundColor = "#f4f4f4";
    arreglodispositivo.push(dispositivo);
```

```
        socket.emit('datos', arreglodispositivo);
                arreglodispositivo = [];
                        }

id_2.addEventListener("click", function () {
                var dispositivo = {};
                        ...
                        ...
                        ...
    arreglodispositivo.push(dispositivo);
      socket.emit('datos', arreglodispositivo);
                arreglodispositivo = [];
                        }
```

De la misma manera que el fichero **app.js** si algún evento enviado desde el sistema físico, el aplicativo se pone en funcionamiento y se debe interpretar mediante la actuación del evento en la página web, la herramienta **socket.on** contendrá una función de datos, datos que deben ser asociados con los diferentes objetos correspondientes y compartidos a todos los clientes con la función `socket.broadcast.emit`.

```
socket.on("datos", function (data) {
       var sizedatos = data.length;
        for (i = 0; i < sizedatos; i++) {
         if (data[i].objeto_1 != null) {
               id_1.disabled = true;
       id_2.style.display = "none";
 id_3.style.backgroundColor = "#f4f4f4";
                        }
        if (data[i].objeto_2  != null) {
               ...
               ...
                        }
          }
  socket.broadcast.emit('datos', data);
  });
```

6.4.2 Implementación en sistema físico (ARDUINO)

A. Interpretación de los objetos (ArduinoJson)

Para interpretar en la plataforma Arduino los diferentes cambios que se realizó dentro del entorno del servidor y el sitio web, nos basaremos en un asistente de ArduinoJson, que permite la interpretación de los objetos y su valor que llega en la cadena de caracteres por el puerto serial del Arduino comunicación denominada UART, cadena que es enviada desde el microprocesador Raspberry pi.
Para esto se debe recordar los nombres de los objetos y el tipo de valor que se indicó en la Tabla 9. En el sitio web https://arduinojson.org/, cuenta con una ventana de asistencia para crear el buffer de salida y entrada.
En la sección INPUT de este asistente se debe ingresar los elementos según la Sintaxis basado en Json:

```
{
"objeto_1": "gps",
"objeto_2": 1,
"objeto_3": false
}
```

El asistente refleja resultados del tamaño del Buffer (JsonBuffer size), **JSON_OBJECT_SIZE(3);** indicando el tamaño del buffer puede entrar o salir 3 objetos:
En este proyecto se debe ingresar los datos correspondientes al objeto y su tipo de valor dentro del arreglo de Json. Tal como se muestra en la siguiente sintaxis de la aplicación.

```
{
"dispositivo":1,
"modofunc":true,
"motor":255,
"presencia":true,
"estado":0,
"imagenprocesada":true,
"clasificador":true,
"sentidomotor":1,
"direcclasificacion":1,
"capturarengrane":false
}
```

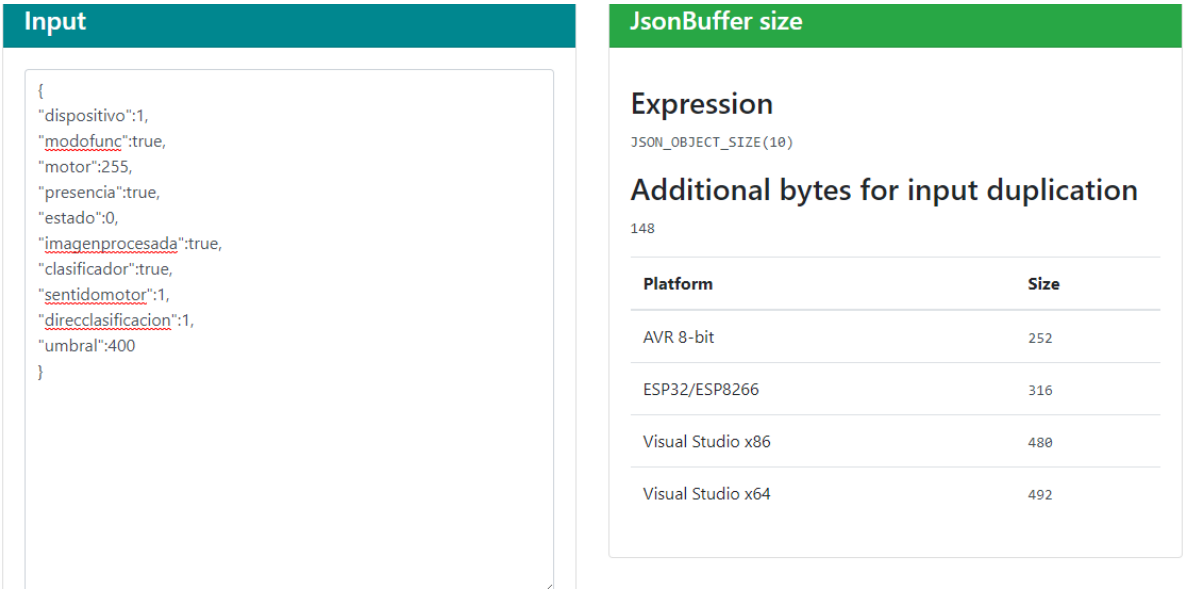


Figura 44. Asistente Arduinojson.org.

En la Figura 44 se muestra el resultado del tamaño de los objetos de Json, adicionalmente muestra el tamaño de bytes que en las diferentes plataformas que se puede utilizar ArduinoJson. En nuestro caso la plataforma a utilizar es AVR 8-bit dentro del grupo ATMEGA, el tamaño que el microcontrolador utilizará es 263 bytes de un total de 32Kbytes.

Para desarrollar el fichero del Arduino(.ino), el primer paso es incluir la librería ArduinoJson.h, que facilitará leer la cadena de caracteres y asignar los diferentes objetos y sus valores correspondientes a cada uno. Luego de declarar variables que ayudara a la programación de los actuadores o sensores, se debe abrir comunicación en el void setup a una velocidad de 115200 baudios, y asignar si las variables son de tipo GPIO e inicializar las diferentes variables auxiliares. El siguiente paso es el void loop, campo para programar las diferentes acciones de los sensores y actuadores.
Dicho esto, seguiremos la siguiente sintaxis según el ejemplo presentado para el encendido/apagado de un led desde el sistema físico o desde el sitio web.

```
#include "ArduinoJson.h"
int inicio = 7;
int paro = 9;
int ledstate = 8;
```

Lo primero es declarar el tamaño de los objetos “bufferSize”, para este ejemplo el tamaño del Objeto en JSON es dos, a continuación, se declara un buffer dinámico para los datos de salida “Buffersalida”, escribiendo los objetos dentro de la variable “bufferSize”. Por último, declaramos la variable “datosalidajson”, variable que guardado el objeto y su valor creado luego de un evento que se obtuvo en el sistema físico.

```
const size_t bufferSize =JSON_OBJECT_SIZE(2);
DynamicJsonBuffer Buffersalida(bufferSize);
JsonObject& datosalidajson = Buffersalida.createObject();
```

Adicional se debe tomar en cuenta las diferentes variables auxiliares a utilizar en el código de programación.

```
unsigned long tiempoinicio = 0;
char d;
int flag = 0;
char serialequipo2=0;
String datospuertoserial = "";
bool haycadena = false;
bool iniciocadena = false, fincadena = false;
```

Inicializamos el Void setup. En este campo se debe abrir la comunicación por el puerto serial a velocidad (115200 baudios) para el envió y recepción de la cadena de caracteres, e indicar las diferentes variables GPIO de entradas o salidas.

```
void setup() {
  Serial.begin(115200);
  pinMode(ledstate, OUTPUT);
  pinMode(inicio, INPUT);
  pinMode(paro, INPUT);
  }
```

Y por último la programación correspondiente dentro de un void loop. En este campo se deberá leer la cadena de caracteres y además de realizar las diferentes acciones de control con el prototipo. El primer paso será verificar si la cadena de caracteres llego forma completa, luego de haber sido guardado en la variable (d) tipo char, mendiante la identificación de los caracteres “{“ y “}” indicando el inicio y final de la cadena respectivamente, así logrando contar con todos los objetos para el funcionamiento correcto del proyecto.

```
void loop() {
    while (Serial.available() > 0) {
      haycadena = true;
      d = Serial.read();
      if (d == '{') {
        datospuertoserial = "";
        iniciocadena = true;
      }
      if (iniciocadena) {
        datospuertoserial.concat(d);
      }
      if (d == '}') {
        fincadena = true;
        break;
      }
    }
```

Luego de que haya comprobado si la cadena ha llegado completamente se procede a leer y asignar los objetos con su valor en diferentes variables. Con la ayuda de la herramienta `String(kv.key).compareTo(“ ”)`.

```
DynamicJsonBuffer Bufferentrada(bufferSize);
      JsonObject&datojson=
Bufferentrada.parseObject(datospuertoserial);
      if (datojson.success()) {
        for (auto kv : datojson) {
          if (String(kv.key).compareTo("led") == true) {
            if (kv.value.as<bool>() == true) {
                        digitalWrite(ledstate, HIGH);
                       // encendido el led
            } else {
                      digitalWrite(ledstate, LOW);
                        // led apagado
             }
            }
           }
         } else {
                  Serial.print(datospuertoserial);
         }
                   datospuertoserial = "";
        }
```

Se ha visto el control de encendido/apagado de un led desde el sitio web. Ahora bien, si el usuario desea interactuar desde el prototipo y verificar los cambios en el sitio web. En el campo de programación del Arduino se debe enviar la cadena caracteres con los objetos correspondientes con la herramienta `datosalidajson.printTo(Serial)`. Previamente guardado los diferentes objetos en la variable `datosalidajson`.

```
if (digitalRead(inicio) == HIGH) {
  flag = 1;
  delay(120);

  if (flag != 0) {
      // la función datossalidajson con el nombre del objeto
      // y su valor
     datosalidajson["led"] = true;
    datosalidajson.printTo(Serial);
    digitalWrite(ledstate, HIGH);
    flag = 0;
  }
}
if (digitalRead(paro) == HIGH) {
  flag = 1;
  delay(120);
  if (flag != 0) {
    datosalidajson["led"] = false;
    datosalidajson.printTo(Serial);
    digitalWrite(ledstate, LOW);
    flag = 0;
  }
}
```

Con el ejemplo ilustrado en los apartados anteriores, permitirá al estudiante conocer la estructura para leer y enviar los diferentes objetos desde el aplicativo web hacia el sistema físico o viceversa.

En la aplicación del prototipo SECVIA, se utilizará las diferentes herramientas expuestas en el ejemplo anterior e identificar los objetos que envíen, reciban o bidireccional por el puerto serial, tal como muestra en la Tabla 11. Donde el sistema podrá trabajar en los modos: automático y manual.

Ítem	Objetos	Prototipo Envió por el puerto serial.	Aplicación Lee los objetos del puerto serial de entrada.	Valor	Funcionalidad
1	dispositivo:		dispositivo:	Int(1)	Indica al prototipo correspondiente.
2	modofunc:		modofunc:	bool(true)	Activación modo automático
				bool(false)	Activación modo manual.
3	motor:		motor:	Int (0-125)	Control de la velocidad del motor
4	presencia:	presencia:		Bool(true/false)	Detección de engranaje
5	estado:	estado:	estado:	Int(-1)	Paro de emergencia del prototipo
				Int(0)	Stanby del prototipo
				Int(1)	Sistema en funcionamiento
6	imagenprocesada:		imagenprocesada:	bool(true/false)	Procesado de engranaje
7	clasificador:		clasificador:	bool(true/false)	Clasificación de engranajes buenos y defectuosos.
8	sentidomotor:		sentidomotor:	Int(-1)	Motor hacia delante
				Int(0)	Motor parado
				Int(1)	Motor hacia atrás
9	direcclasificacion:		direcclasificacion:	Int(-1)	Servomotor dirección engranajes defectuosos.
				Int(0)	Movimiento en Stanby
				Int(1)	Servomotor dirección engranajes correctos.
10	umbral:		umbral:	Int(400)	Valor para la calibración del sensor de posición
11	capturarengrane:		capturarengrane:	bool(true/false)	Activa iluminación.

Tabla 11. Objetos utilizados en el prototipo SECVIA.

En la Figura 45, se muestra el algoritmo que se debe seguir para la implementación del modo manual en el prototipo SECVIA, el algoritmo debe funcionar cuando el modo de funcionamiento es falso. Este valor deberá ser enviado desde aplicativo web e interpretado por el Arduino y trabajar en este campo.

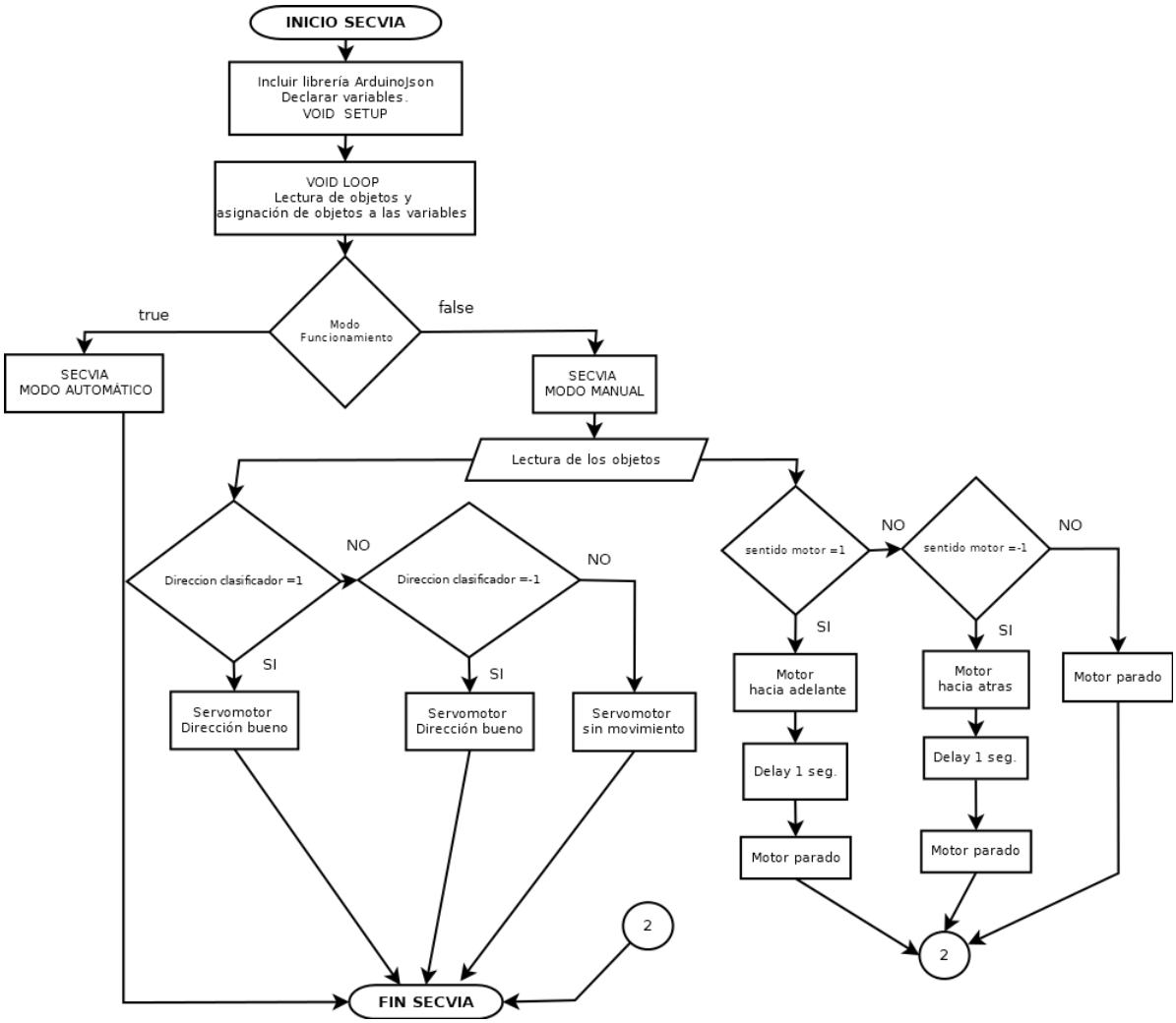


Figura 45. Funcionamiento del prototipo en modo manual.

Por último, se debe conocer los respectvos pines a utilizar para la asignación de los diferentes actuadores del prototipo. En el anexo A, se muestra el diagrama correspondiente y el diseño de la PCB a utilizar como shield en el Arduino Uno, para ello en la Tabla 12, se indica los diferentes puertos/pines para la programación de los actuadores, sensores y comandos de control.

GPIO	Denomin.	Asinación:	Tipo	Funcionalidad
Pin #6	red	OUTPUT	PWM	Combinación de valores para obtener el color deseado en el Diodo Led RGB.
Pin #3	blue	OUTPUT	PWM	
Pin #5	green	OUTPUT	PWM	
Pin #13	objbueno	OUTPUT	Digital	Led indicador para la clasificación de engranajes buenos.
Pin #8	objmalo	OUTPUT	Digital	Led indicador para la clasificación de engranajes malos.
Pin #A0	receptor	INPUT	Analógico	Obtiene el valor del sensor de posición cuando hay un corte del haz de Infrarrojo.
Pin #9	emerg	INPUT	Digital (bool)	Comando físico para el paro emergencia del prototipo
Pin #7	start	INPUT	Digital (bool)	Inicio/paro del sistema automático del prototipo

Pin #4	ledcamara	INPUT	Digital (bool)	Encendido de la iluminación para la captura de la imagen.
Pin #10	pinservo	OUTPUT	PWM (int)	Selección de engranajes bueno y malos.
Pin #11	motorpwm	OUTPUT	PWM (int)	Velocidad del motor para la cinta de transporte.
Pin #12	motordir	OUTPUT	Digital (bool)	Sentido de giro de la cinta de transporte

Tabla 12. Asignación de GPIOs para el ARDUINO UNO

6.5 Guía de laboratorio #2. Calibración del sensor de posición.

En la Guía de laboratorio #1 se logró interactuar con los diferentes actuadores que cuenta el sistema CPS académico. En esta segunda parte de la guía práctica se tratará de comprender sobre los distintos sensores que puede contar un equipo, para nuestro prototipo haremos referencia específicamente al sensor de posición.

La estructura del sensor de posición cuenta con un diodo led emisor y un diodo receptor, el led emisor envía haz de luz a una distancia cierta siendo capturada por el diodo receptor, generando un valor de umbral con relación a la distancia y posición del emisor, este valor debe ser calibrado para que exista la detección de objetos.

Considerando lo aprendido en la guía de Laboratorio #1. El primer paso es añadir el comando que servirá para la calibración del umbral del sensor posición.
En el subsitio de control y monitorización, se debe crear una nueva sección dentro del fichero controlymoni.html. Esta sección puede estar ubicado en cualquier posición del interfaz del subsitio, recomendación la sección debe estar al final de la sección del encabezado. En el espacio de esta sección para nuestra aplicación será crearemos un comando deslizante(slider). Este comando deslizante se crea dentro de una división de la sección y detallando la forma(`<form>`) de este comando, consiguiente de una entrada **input**, especificado su id, el tipo(`range`), su máximo y su mínimo valor de rango.
Para esto utilizaremos la siguiente sintaxis ya desarrollada:

```
<section id="showcase2" style="background-color:#268982"
style="margin-left: 20px" style="margin-right: 20px">
  <div class ="container" style="font-size: 12px">
            <h2>CALIBRACION DEL SENSOR DE POSICIÓN</h2>
  </div>
  <div class ="container">
    <form>
                  <p>  Valor de Umbral </p>
        <input id="umbral" type="range" name="sliderumbral"
               min="800" max="1200" value="800">
    </form>
          <label id="umbralvalor" style="font-size: 15px"
       style="text-align: center" style="color: #000000">
    </label>
  </div>
</section>
```

Creado esta sección destinado a la calibración del umbral de un sensor de posición, en la Figura 46 se muestra el interfaz que se ha generado y que se contará en la aplicación web.

Figura 46. Comando para la calibración del sensor de posición.

Hay que recordar que cada comando cuenta con sus respectivos identificadores, con la final de con el entorno JavaScript del servidor y de la aplicación web. En esta parte de la guía práctica se utilizará los diferentes identificadores presentados en la Tabla 13 tanto para el botón, como para los textos.

Tipo	Nombre del Slider	Identificador del botón (id)	Identificador del texto (id)
input	CALIBRACIÓN DEL SENSOR DE POSICIÓN	id="umbral"	Id="umbralvalor"

Tabla 13. Identificador del comando para la calibración del sensor.

Ahora, se deberá realizar los diferentes eventos dinámicos en el entorno Javascript (main.js), y poder añadir eventos después que se haya interactuado con este comando, enviando él o los identificadores y sus valores para ser enviado al servidor. Para obtener y guardar la información del id, se utiliza la herramienta `var id=document.getElementById("id").`

Con la herramienta `elemento.addEventListener()`, leeremos los eventos del identificador del sitio web, para luego asignarle alguna actividad y enviar al sistema físico una vez que este elemento se haya guardado en un vector `arreglodispositivo.`

```
umbral.addEventListener("click", function () {
    var dispositivo = {};
    dispositivo.dispositivo = 1;
    dispositivo.umbral = Number(umbral.value);
    umbralvalor.innerHTML = umbral.value;
    arreglodispositivo.push(dispositivo);
    console.log(arreglodispositivo);
    socket.emit('datos', arreglodispositivo);
    arreglodispositivo = [];
});
```

Para la obtención del estado del cliente se utilizará la herramienta `socket.on`, este estado guardará el valor de este objeto que viene del sitio web.

```
if (data[i].umbral != null) {
        umbral.value = Number(data[i].umbral);
        umbralvalor = Number(data[i].umbral);
    }
```

En el fichero **app.js,** fichero correspondiente al servidor. En el servidor también debe incluirse el nuevo objeto de esta práctica. Cabe recordar que el nuevo objeto debe de estar dentro del arreglo Json **dispositivo.**

Para establecer la conexión y leer las variables del arreglo Json se utiliza la herramienta **io.on,** estas variables deberán ser agregados en un vector (`arreglodispositivo`) mediante el método **push**, y al final emitiendo todo este vector a la conexión del servidor.

Finalizaremos con la plataforma de Arduino, aquí se debe incrementar el tamaño del buffer a 11 objetos: `JSON_OBJECT_SIZE(11),` y se aplicará los mismos procedimientos de programación de la guía práctica #1 para su correspondiente funcionamiento.

El funcionamiento se muestra mediante el algoritmo presentado en la Figura 47, el mismo que servirá solo para habilitar el sistema del mando manual, sistema que fue implementado en la guía práctica # 1.

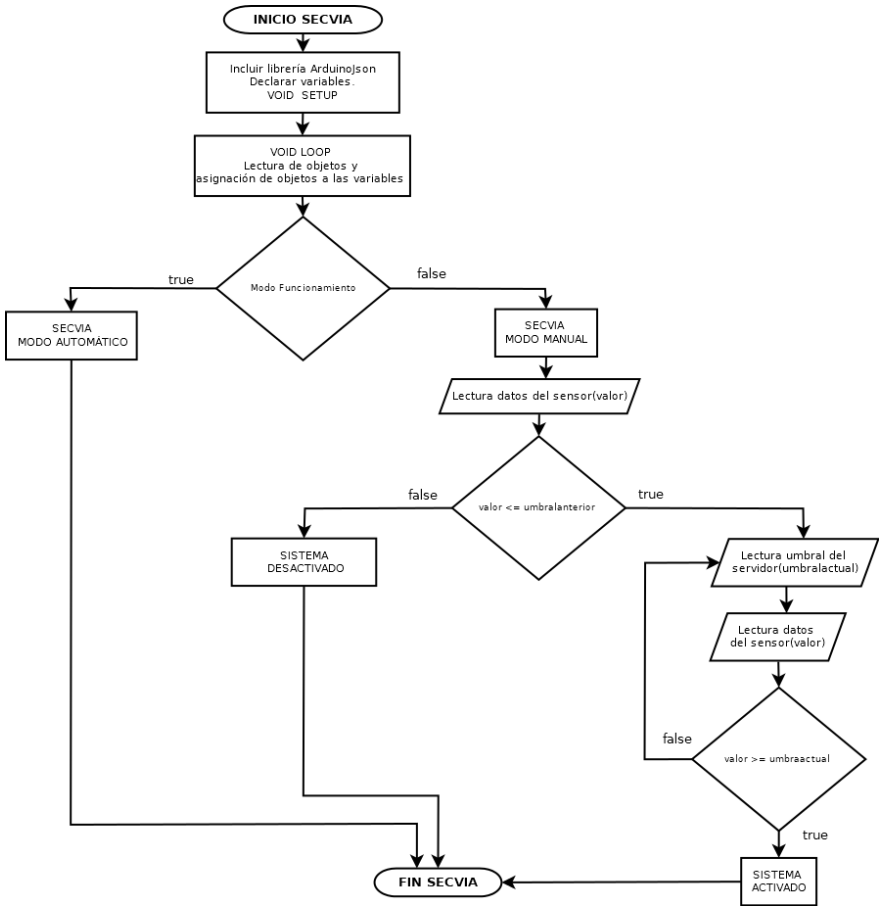


Figura 47. Algoritmo Sistema manual y calibración sensor

6.6 Recomendaciones para las comprobaciones del estado del servidor web y del clientesocket

En este apartado se presentará algunos comandos necesarios que servirá para comprobar, inicializar o paralizar los diferentes motores de ejecución tanto el servidor web como el cliente correspondiente al prototipo. Comandos que ayudará al estudiante a verificar si el sistema se encuentra en error.

El principal comando `systemctl` es un ejecutador que ayuda al servidor web como al cliente funcionen automáticamente, sin la necesidad de que el usuario ejecute manualmente los correspondientes interfaces mencionados.

6.6.1 Servidor

- **Iniciar un servidor.**

La siguiente línea de comando presentando a continuación permitirá al sistema ejecutar.

```
sudo systemctl start servidorweb
```

- **Parar un servidor.**

Ahora bien si el estudiante desea apagar o parar al servidor, para realizar cualquier cambio o modificación en los ficheros correspondientes al servidor.

```
sudo systemctl stop servidorweb
```

- **Reiniciar un servidor.**

Si el usuario o estudiante observa que el servidor presenta alguna anomalía, con el comando siguiente permitirá reiniciar al servidor web y por ende reinicia a los valores de los clientes conecatados a este servidor.

```
sudo systemctl restart servidorweb
```

- **Estado del servidor.**

El último comando presentado permitirá al usuario ver en que estado se encuentra el servidor web, es decir si el servidor está en funcionamineto o presente algún problema.

```
sudo systemctl status servidorweb
```

6.6.2 Cliente Socket

De la misma manera que el servidor web, el motor ejecutador del cliente del prototipo, se utilizará los mismos comandos, haciendo referencia al fichero ejecutador del cliente. Tal como se muestra acontinuación:

```
sudo systemctl start programabanda
sudo systemctl stop programabanda
sudo systemctl restart programabanda
sudo systemctl status programabanda
```

Por último, si el estudiante desea ejecutar manualmente cada vez que ponga en funcionamiento el prototipo puede utilizar los siguientes comandos.

- **Motor ejecutador Servidor Web.**

```
cd Documentos/UPC_TFM_Fernando_Jacome/ServidorWeb/bin
sudo ./www
```

- **Motor ejecutador Cliente Socket SECVIA.**

```
cd Documentos/UPC_TFM_Fernando_Jacome/ClienteSocket/dist/Debug/GNU-Linux
sudo ./clientesocket
```

7. CONCLUSIONES

En este capítulo se presenta una visión global de los resultados del trabajo. Para ello partiremos de los diferentes objetivos marcados en un inicio, evaluándolos uno a uno, con la finalidad de verificar si se ha cumplido en su totalidad o parcialmente. Tras el análisis de los objetivos se plantearán las conclusiones y, por último, los posibles trabajos futuros de actualización y mejoras.

El objetivo principal ha sido la creación de un entorno ciber-físico de nivel académico, que permita la ejecución de un cierto sistema físico desde un sitio o aplicativo web. Durante el desarrollo del capítulo 4, se ha podido ver cómo se diseñó la parte mecánica y electrónica del sistema físico. Se han integrado sensores y actuadores y se han utilizado microcontroladores y un microprocesador de bajo coste. En el capítulo 5 se comprueba que el primer objetivo principal se ha cumplido completamente mediante la implementación de un prototipo denominado SECVIA. Está compuesto por un sistema de banda de transporte, un sistema infrarrojo para detección de engranajes, una cámara para el sistema de visión y posterior procesado de la imagen capturada y, por último, el sistema de clasificación de los objetos en función al resultado obtenido del procesado de la imagen.

Para dar cumplimiento a este objetivo principal se establecieron algunos objetivos específicos. Estos objetivos se enfocaron hacia la integración de sensores y actuadores, la creación de un interfaz de comunicación para un CPS, capacidad de monitorear, coordinar y controlar en tiempo real desde un aplicativo web a un sistema físico y, por último, procesar y clasificar engranajes en función de su calidad a través de un sistema de visión. En el capítulo 4, se ha podido verificar la implementación de los dispositivos hardware y programas software. Además, se han desarrollado los algoritmos necesarios para la comunicación del CPS, aplicando el mecanismo de comunicación servidor-cliente dentro de un aplicativo web y el algoritmo para el procesado de un engranaje recto luego de obtener la imagen desde un dispositivo de visión. En el capítulo 5, correspondiente a resultados de este trabajo, se ha logrado validar en su totalidad los objetivos específicos luego de implementar y desarrollar el prototipo.

Por otro lado, el prototipo SECVIA se orientó al aprendizaje de los CPS. Para ello surgió la necesidad de desarrollar una guía práctica, que describe la implementación de un sistema de control manual de SECVIA y la integración de un método de calibración del umbral de detección de objetos. Se plantearon una serie de objetivos para el aprendizaje de un sistema CPS, mediante la interacción con los diferentes actuadores y sensores del prototipo de forma manual.

La guía práctica desarrollada permitirá al alumno implementar funciones y procedimientos adicionales en el prototipo, siendo una estrategia didáctica que contribuye a la enseñanza-aprendizaje de los CPS. De este modo, el estudiante tendrá la posibilidad de corroborar y comprobar principios y conceptos de los CPS en la industria, enfrentándose y poniendo a prueba sus habilidades en estos sistemas.

La implementación del prototipo SECVIA presta los beneficios de disponer de una mayor capacidad de interconexión mediante la interoperabilidad con la nube y redes inalámbricas en múltiples plataformas computacionales, a traves de un interfaz gráfico entre usuarios y el sistema por medio de un aplicativo web.

7.1 Líneas Futuras.

Si bien, es posible ampliar el estudio de los CPS realizado en este proyecto, existe una multitud de direcciones que podrían mejorarlo. En este punto se ofrecen opciones de mejora tanto de software como de hardware.

- **Software.**

Implementar una base de datos en la nube, herramienta importante para una industria debido a que permite organizar de manera óptima los datos recogidos del sistema físico y tenerlos fácilmente accesibles en cada momento que sean solicitados. Asimismo, permite reducir costos en equipos de cómputo para el almacenamiento.

Implementar un servidor en la nube tendría ventajas de flexibilidad para acceder a recursos cuando sea requerido, mayor rentabilidad y confiabilidad que los servidores tradicionales, debido a que en la nube no tienden a suspender el servidor.

Añadir algortimos de visión y procesamiento de otro tipo de objetos que no sean engranajes rectos. De este modo se dispondra de un sistema de clasificación de varios objetos en el CPS.

Insertar algoritmos con metodologías Maching Learning, técnicas desarrollas con la computación como base, que permiten a las máquinas aprender. El sistema puede evolucionar constantemente y se nutre de nuevos datos e información adquiridos en cada paso. El objetivo final es que el sistema vaya perfeccionando su comportamiento de forma autónoma y logre un nivel de eficiencia óptimo.

- **Hardware**

Integrar dispositivos de control, actuadores y sensores adicionales, con el propósito de lograr contar con un sistema de mayor complejidad, más cercano al ámbito industrial.

Utilizar un dispositivo programable que sea de uso alternativo al Raspberry pi y Arduino, se prodría pensar en una FPGA. Este dispositivo puede ser capaz de integrar la parte hardware y software.

Este estudio sobre CPS también abre una ventana de lineas de trabajo futuro en relación a la enseñanza de estos sistemas.

Con el objetivo de ampliar y mejorar la guía práctica del alumno, con aplicaciones nuevas que permitan al estudiante interactuar más fácilmente con el prototipo SECVIA.

8. AGRADECIMIENTOS.

Al finalizar una etapa más en mi vida y luego de realizar un trabajo arduo y lleno de dificultades como lo es el desarrollo de este proyecto, primeramente, me gustía agradecerle a Dios por la bendición para llegar a cumplir el sueño anhelado de un Máster Universitario.

También quiero agradecer a mi padre Mentor y mi madre Gladys, que a pesar la distancia supieron darme su apoyo incondicional en cada momento, a mis hermanos José y María Teresa que siempre han estado en buenos y malos momentos.

A mi tutor del TFM Jordi Prat, quién ha sido un soporte fundamental para el desarrollo de este proyecto, por medio de su paciencia y sus continuas revisiones del progreso de la implementación de este prototipo.

A mis tíos, tías, primos y primas por apoyarme y animarme a la distancia a lograr este sueño que se esta haciendo realidad.

A la familia Barrionuevo Mejía, quienes por su colaboración y amabilidad supieron acogerme en su familia durante mi estancia en Barcelona.

A mis amigos, por ser parte de mi vida, de momentos triste y alegres, por apoyarme, por nunca dejarme caer, por estar siempre ahí cuando más se lo requieren.

Bibliografía.

[1] E. Gomez, «smartLIGHTING,» 17 Mayo 2017. [En línea]. Available: https://smart-lighting.es/iot-ciberfisicos-industria/.

[2] D. S. D. María Antolín Fernández, Abril 2016. [En línea]. Available: www.coiicv.org.

[3] Dr. Werner Steinhögl, European Commision, «ICT 1 – Smart Cyber-Physical Systems in H2020,» 23 Octubre 2013. [En línea]. Available: https://ec.europa.eu/newsroom/index.cfm.

[4] Copyright © 2018 by LecLife.com, "Tejas Dubhir @ smart city model.," [Online]. Available: https://www.leclife.com/index.php?alec=search&qlec=Tejas%2520Dubhir%2520%2540%2520smart%2520city%2520model..

[5] J. Rivera, «DOMÓTICA CON RASPBERRY PI Y ANDROID EN TIEMPO REAL,» 17 Marzo 2016. [En línea]. Available: http://riverajefer.blogspot.com/2016/03/domotica-con-raspberry-pi-y-android-en.html.

[6] FESTO, "Estación de Clasificación," [Online]. Available: https://www.festo-didactic.com/es-es/productos/mps-sistema-de-produccion-modular/estaciones/estacion-de-clasificacion-final.htm?fbid=ZXMuZXMuNTQ3LjE0LjE4LjYwNi4zOTQ4.

[7] Karen Rose, Scott Eldridge, Lyman Chapin, «INTERNET SOCIETY,» Octubre 2015. [En línea]. Available: https://www.internetsociety.org/wp-content/uploads/2017/09/report-InternetOfThings-20160817-es-1.pdf.

[8] M. B. Andrés, «INTERNET DE LAS COSAS,» 2018. [En línea]. Available: https://www.editorialreus.es/static/pdf/primeraspaginas_9788429020380_internetdelascosas.pdf.

[9] F. N. S. NSF, «Cyber-Physical Systems (CPS),» Mayo 2015. [En línea]. Available: https://www.nsf.gov/pubs/2015/nsf15541/nsf15541.pdf.

[10] European Commission, «Research and Innovation,» [En línea]. Available: http://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/.

[11] ECSEL, «Adopting the E CSE L Multiannual Strategic Plan,» Julio 2014. [En línea]. Available: http://ec.europa.eu/research/participants/data/ref/h2020/other/legal_basis/jtis/ecsel/ecsel-multi-stratplan_en.pdf.

[12] NIST UC Berkeley , "Ptolemy Project," 2012. [Online]. Available: https://ptolemy.berkeley.edu/projects/cps/.

[13] NEO (Networking and Emerging Optimization), «Networking and Emerging Optimization,» [En línea]. Available: http://neo.lcc.uma.es/evirtual/cdd/tutorial/Indice.html.

[14] culturacion, "Cuál es la utilidad de las aplicaciones cliente/servidor," [Online]. Available: http://culturacion.com/cual-es-la-utilidad-de-las-aplicaciones-clienteservidor/.

[15] C. R. León, «Websockets,» Junio 2013. [En línea]. Available: http://nereida.deioc.ull.es/~stw/perlexamples/node26.html.

[16] websocket.org, «About HTML5 WebSocket,» [En línea]. Available: https://www.websocket.org/aboutwebsocket.html.

[17] node js,Ryan Lienhart Dahl, «Ryan Lienhart Dahl,» Mayo 2009. [En línea]. Available: https://es.wikipedia.org/wiki/Node.js.

[18] hipertextual, «Socket.io: Comunicación bidireccional en tiempo real para JavaScript,» 2014. [En línea]. Available: https://hipertextual.com/archivo/2014/08/socketio-javascript/.

[19] codeburst.io, «Real Time Web App | React.js + Express + Socket.io,» Octubre 2017. [En línea]. Available: https://codeburst.io/isomorphic-web-app-react-js-express-socket-io-e2f03a469cd3.

[20] JFernandez, «Node.js + Socket.io = Real-Time IO.,» Diciembre 2012. [En línea]. Available: https://blogs.igalia.com/jfernandez/2012/12/19/node-js-socket-io-real-time-io/.

[21] EUATM, Sección Informática , "INTRODUCCIÓN AL WEB," [Online]. Available: http://www.edificacion.upm.es/informatica/documentos/www.pdf.

[22] J. Alonso, "Editorial Universidad de Sevilla," Mayo 2005. [Online]. Available: http://institucional.us.es/revistas/comunicacion/5/07alonso.pdf.

[23] A. Sánchez, "Introducción a las páginas web dinámicas," [Online]. Available: http://docplayer.es/6719047-Lenguaje-html-2-introduccion-a-las-paginas-web-dinamicas-alex-sanchez.html.

[24] MDN web docs, «¿Qué es JavaScript?,» Septiembre 2018. [En línea]. Available: https://developer.mozilla.org/es/docs/Learn/JavaScript/First_steps/Qu%C3%A9_es_JavaScript.

[25] Yakin15, hamfree, wacoloo35, javierpolit, «MDN web docs,» Marzo 2018. [En línea]. Available: https://developer.mozilla.org/es/docs/Learn/CSS/Introduction_to_CSS/Como_funciona_CSS.

[26] json.org, "Introducción a Json," 1999. [Online]. Available: https://www.json.org/json-es.html.

[27] Y. Sharma, «%{confusedCoders},» 15 Octubre 2016. [En línea]. Available: https://confusedcoders.com/general-programming/go-lang/create-a-basic-distributed-system-in-go-lang-part-2-http-server-json-requestresponse.

[28] B. B. ARDUINOJSON, Mastering ArduinoJson Efficient JSON serialization for embedded C++.

[29] OpenCV, Intel, "Open Source Computer Vision," 2000. [Online]. Available: https://docs.opencv.org/3.4.1/d1/dfb/intro.html.

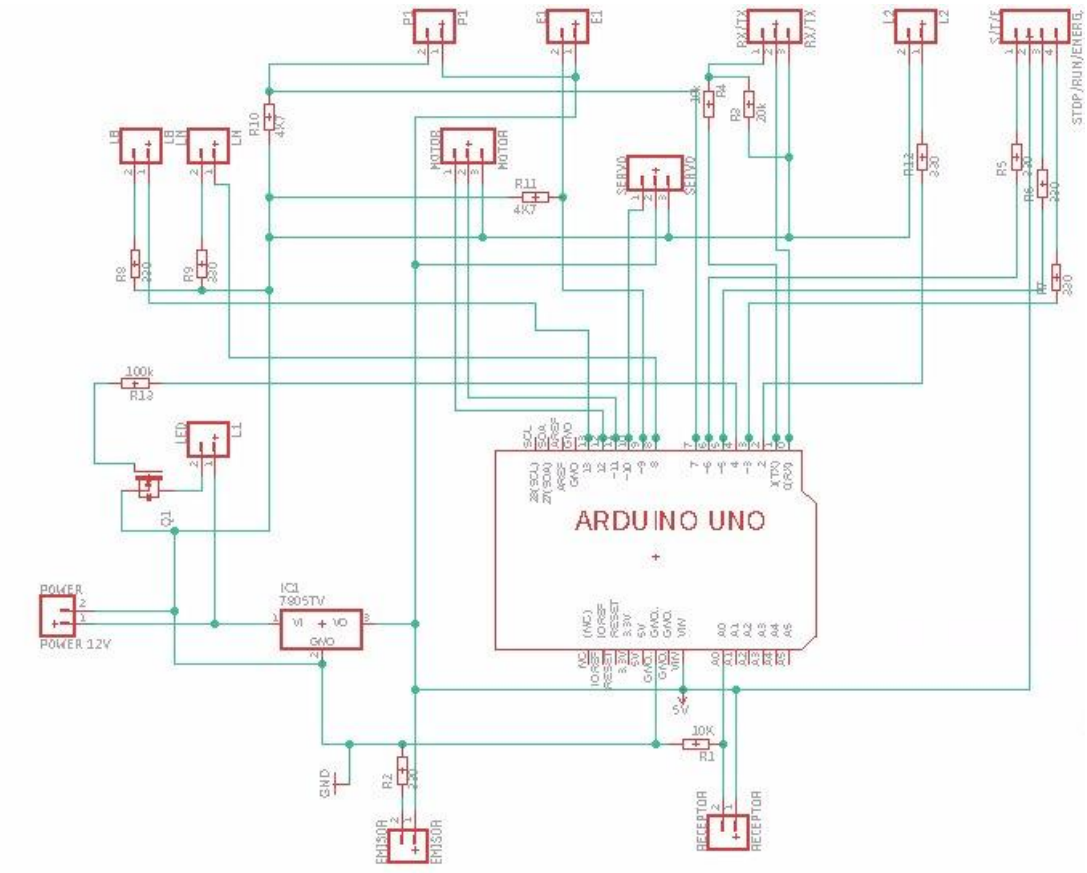
[30] J. A. Taquía Gutiérrez, Octubre 2017. [En línea]. Available: https://dialnet.unirioja.es/descarga/articulo/6230450.pdf.

[31] github, "A fast JSON parser/generator for C++ with both SAX/DOM style API," github, 13 Noviembre 2011. [Online]. Available: https://github.com/Tencent/rapidjson/. [Accessed 07 Octubre 2018].

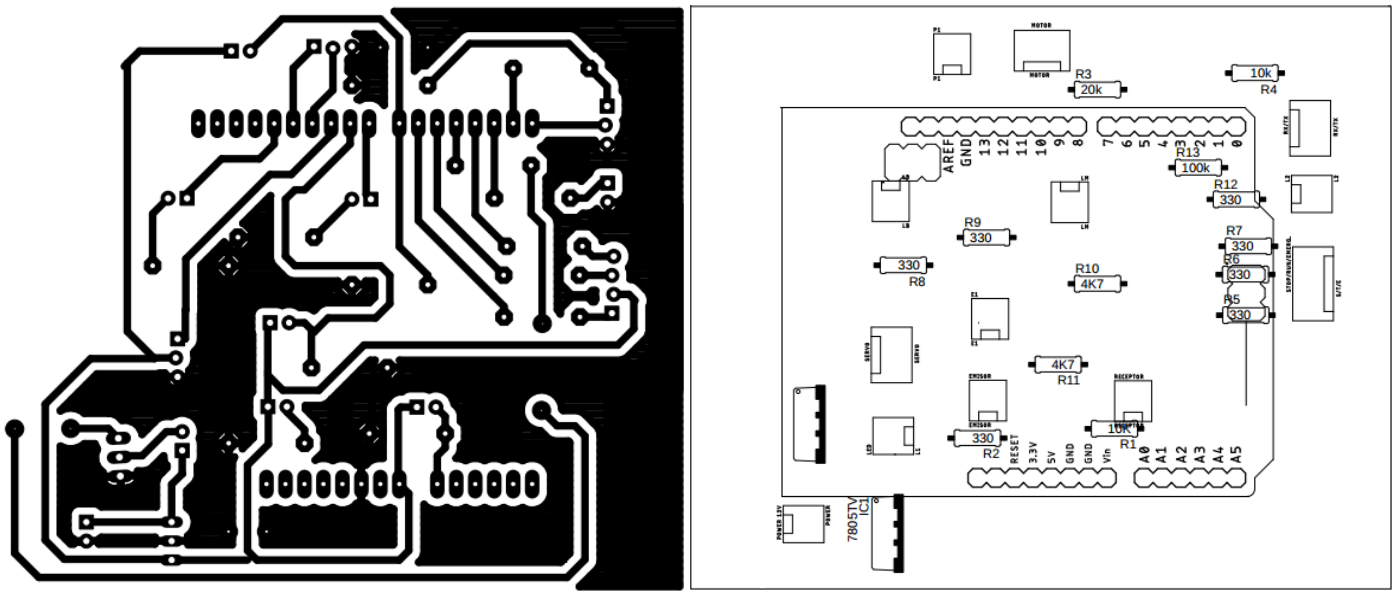
[32] github, "C++ websocket client/server library," github, 04 Septiembre 2011. [Online]. Available: https://github.com/zaphoyd/websocketpp. [Accessed 16 Julio 2018].

ANEXOS.

ANEXO 1 - DISEÑO ELECTRÓNICO.



Circuito para el shield de Arduino



Diseño PCB del prototipo

ANEXO 2 – ESTUDIO ECÓNOMICO.

Estudio Económico de Producción SECVIA.	
COSTOS DIRECTOS (1) +(2)	**TOTAL**
Componentes y dispositivos electrónicos (1)	193,35
Elementos mecánicos de construcción (2)	168,00
COSTOS INDIRECTOS	
Materiales indirectos.	50,00
Costos de Mano de Obra (3)	27840,00
COSTO TOTAL	**EUR 28253,35**

ÍTEM	COSTOS DIRECTOS (1) +(2)	CANTIDAD	COSTO TOTAL
COMPONENTES Y DISPOSITIVOS ELECTRÓNICOS			
1	Resistencia 330Ω	4	0,80
2	Resistencia 4,7 kΩ	2	0,50
3	Regulador de tensión LM7805	1	0,51
4	Conectores MOLEX 4X1 Macho/Hembra	1	0,20
5	Conectores MOLEX 3X1 Macho/Hembra	3	0,60
6	Conectores MOLEX 2X1 Macho/Hembra	10	2,00
7	Pin macho Shield Arduino	24	2,00
8	Driver de Motor DC	1	30,00
9	Arduino UNO	1	15,00
10	Tarjeta Raspberry PI 3B+	1	39,33
11	MOSFET IRF530N	1	1,30
12	Servo Motor 5V.	1	7,00
13	Motor DC 12V.	1	30,00
14	Pulsador Normalmente Abierto	1	1,50
15	Botón tipo Hongo Emergencia	1	3,00
16	Diodo LED RGB	1	1,50
17	Diodo LED	2	0,40
18	Tira de led circular	1	6,00
19	Tarjeta micro USB	1	10,00
20	Diodo Led Emisor	1	0,51
21	Diodo Led Receptor	1	1,20
22	Cámara	1	40,00
		COSTO (1)	**EUR 193,35**
ELEMENTOS MECÁNICOS DE CONSTRUCCIÓN			
23	Perfil CNC 20x20	3mts	24,00
24	Rodillos para cinta de transporte	2	20,00
25	Ángulos tipo L para perfil CNC 20x20	10	5,00

26	Tornillo M5*10*6 para CNC	50	5,00
27	Tuercas M5 para Tornillo CNC	50	4,00
28	Cinta de transporte 10cmx122cm	1	20,00
29	Chumacera Diámetro interno 8mm	4	12,00
30	Correa de Transmisión	1	3,00
31	Polea motriz (motor)	1	30,00
32	Polea conducida (rodillos)	1	10,00
33	Estructura para cinta de transporte	1	10,00
34	Impresiones 3D	1	25,00
		COSTO (2)	**EUR 168,00**

ÍTEM	**COSTE DE MANO OBRA (3).**	**CANTIDAD HORAS**	**COSTE HORA**	**COSTE TOTAL**
1	Investigación de CPS	80 h.	40	3200
2	Desarrollo prototipo SECVIA	160 h.	40	6400
3	Diseño y desarrollo electrónica.	16 h.	40	640
4	Desarrollo de algoritmos	320 h.	40	12800
5	Desarrollo guía práctica.	40 h.	40	1600
6	Pruebas de funcionamiento	80 h.	40	3200
			COSTO TOTAL(3)	**EUR 278740,00**

ANEXO 3 –ALGORITMO SERVIDOR WEB.

- **Fichero index.js**

```
var express = require('express');
var router = express.Router();

/* GET home page. */
router.get('/', function(req, res) {
    console.log("prelogin");
  res.sendfile(__dirname + "/public/index.html");
  console.log("index");
  console.log(path.join(__dirname, 'public'));
});

module.exports = router;
```

- **Fichero app.js**

```
var express = require('express');
var path = require('path');
var favicon = require('static-favicon');
var logger = require('morgan');
var cookieParser = require('cookie-parser
');
var bodyParser = require('body-parser');
var routes = require('./routes/index');
var users = require('./routes/users');
var fs = require('fs'); //require filesys
tem module
var mongo = require('mongodb');     ///Añ
adido base de datos MongoDB
var MongoClient = require('mongodb').Mong
oClient;
var url = "mongodb://localhost:27017/";
var numeroconectados = 0;
var numeroimagenes = 0;
var app = express();
var server = require('http').Server(app);
var io = require('socket.io')(server);
var imagenes = {//Donde guardar las image
nes
    nombre: "",
    nombreprocesada: "",
    fecha: new Date(),
    errores: "",
    numerodientes: 0,
    modulo: 0.0,
    paso: 0.0,
    diametroexterno: 0.0
};
var totalimagenes = [];

var valormotorservidor = 0;
var valorpresenciaservidor = false;
var valorclasificadorservidor = false;
var valorestadoservidor = 0;
var rutaimagen = "/images/Logo_UPC.png";
var rutaimagenprocesada = "";
//INICIO PARTE MANUAL
var modofuncservidor = false;
var sentidomotorservidor = 0;
var direcclasificacionservidor = 0;
var capturarengraneservidor = 0;
var umbralservidor = 0;
//FIN PARTE MANUAL

var dispositivo = {
    dispositivo: 1,
    motor: 255,
    presencia: false,
    estado: -1,
    imagenprocesada: false,
    clasificador: false,
   //INICIO PARTE MANUAL
    //añado la parte manual
    modofunc: false,
    sentidomotor: 1,
    direcclasificacion: 1,
    capturarengrane: false,
    umbral: 100
    //FIN PARTE MANUAL
};

var ledservidor = false;
var equipo2 = {
    equipo2: 1,
    led: false
};
var arreglodispositivo = []; // inicia en
forma de vector
var fechaimagen;
var errores;
var obj;  //Variable para leer JSON

// view engine setup
app.set('views', path.join(__dirname, 'vi
ews'));
app.set('view engine', 'jade');

app.use(favicon());
app.use(logger('dev'));
app.use(bodyParser.json());
app.use(bodyParser.urlencoded());
app.use(cookieParser());
```

```
app.use(require('less-middleware')({src:
path.join(__dirname, 'public')}));
app.use(express.static(path.join(__dirnam
e, 'public')));

app.use('/', routes);
app.use('/users', users);

/// catch 404 and forwarding to error han
dler
app.use(function (req, res, next) {
    var err = new Error('Not Found');
    err.status = 404;
    next(err);
});

/// error handlers

// development error handler
// will print stacktrace
if (app.get('env') === 'development') {
    app.use(function (err, req, res, next
) {
        res.status(err.status || 500);
        res.render('error', {
            message: err.message,
            error: err
        });
    });
}

// production error handler
// no stacktraces leaked to user
app.use(function (err, req, res, next) {
    res.status(err.status || 500);
    res.render('error', {
        message: err.message,
        error: {}
    });
});
//Cargo el JSON si este existe
fs.readFile(__dirname + '/public/images/i
magenes.json', 'utf8', function (err, dat
a) {
    if (!err) {
        console.log("Leyendo");
        obj = JSON.parse(data);
        numeroimagenes = obj.length - 1;
        for (var x = 0; x < obj.length; x
++) {
       totalimagenes.push(obj[x]);
        }
        rutaimagenprocesada = totalimagen
es[totalimagenes.length - 1].nombreproces
ada;
        console.log("Finalizo de leer");
    } else {
        numeroimagenes = 0;
        totalimagenes = [];
        rutaimagenprocesada = rutaimagen;
    }
    console.log("Acabo");
});
io.on('connection', function (socket) {
    numeroconectados += 1;
    console.log("<<Usuario conectado>> Us
uarios en linea: " + numeroconectados);
    if (dispositivo.motor != null) {
  dispositivo.motor = valormotorservidor;
    }
   if (dispositivo.clasificador != null)
{
     dispositivo.clasificador = valorclas
ificadorservidor;
    }
    if (dispositivo.estado != null) {
        dispositivo.estado = valorestados
ervidor;
    }
    if (dispositivo.presencia != null) {
        dispositivo.presencia = valorpres
enciaservidor;
    }

//INICIO PARTE MANUAL
    //CONEXION A LA LIBRERIA SOCKETIO PAR
TE MANUAL
   if (dispositivo.modofunc != null) {
        dispositivo.modofunc = modofuncse
rvidor;
    }
    if (dispositivo.sentidomotor != null)
{
      dispositivo.sentidomotor = sentidom
otorservidor;
    }
    if (dispositivo.direcclasificacion !=
null) {
      dispositivo.direcclasificacion = di
recclasificacionservidor;
    }
    if (dispositivo.capturarengrane != nu
ll) {
        dispositivo.capturarengrane = cap
turarengraneservidor;
    }
    if (dispositivo.umbral != null) {
        dispositivo.umbral = umbralservid
or;
    }
//FIN PARTE MANUAL

    //segundo sistema
    if (equipo2.led != null) {
        equipo2.led = ledservidor;
    }

    arreglodispositivo.push(dispositivo);
    dispositivo = {};
       arreglodispositivo.push(equipo2);
    equipo2 = {};
    socket.emit("nuevousuario", arreglodi
spositivo);
    arreglodispositivo = [];
    socket.emit("lastimagen", rutaimagenp
rocesada);
    socket.on('disconnect', function () {
        numeroconectados -= 1;
        console.log('<<Usuario desconecta
do>> Usuarios en linea: ' + numeroconecta
dos);
```

```
    });

    socket.on("datos", function (data) {
        console.log(data);
        var sizedatos = data.length;
        console.log("Tamaño de datos: " +
sizedatos);
        for (i = 0; i < sizedatos; i++) {
            if (valormotorservidor != nul
l) {
             valormotorservidor = data[i]
.motor;
            }
         if (valorpresenciaservidor != nu
ll) {
                valorpresenciaservidor =
data[i].presencia;
            }
            if (valorclasificadorservidor
!= null) {
                valorclasificadorservidor
= data[i].clasificador;
            }
            if (valorestadoservidor != nu
ll) {
             valorestadoservidor = data[i
].estado;
            }

          //INICIO PARTE MANUAL
            //añado conexion parte manual
            if (modofuncservidor != null)
{
     modofuncservidor = data[i].modofunc;
            }

            if (sentidomotorservidor != n
ull) {
                sentidomotorservidor = da
ta[i].sentidomotor;
            }

           if (direcclasificacionservidor
!= null) {
                direcclasificacionservido
r = data[i].direcclasificacion;
            }

   if (capturarengraneservidor != null) {
               capturarengraneservidor =
data[i].capturarengrane;
            }

        if (umbralservidor != null) {
        umbralservidor = data[i].umbral;
            }
//FIN PARTE MANUAL

 // AÑADO DE UN SEGUNDO SISTEMA SI DESEO
           if (ledservidor != null) {
               ledservidor = data[i].led;
            }
        }
        socket.broadcast.emit('datos', da
ta); //Envia a todos los clientes
        console.log("Enviando datos");
    });

    socket.on('imagen', function (data) {
        console.log("Llego una imagen");
        console.log(data);
     fechaimagen = new Date(data.fecha);
        //errores = data.errores;
        console.log("Llego una imagen cre
ada en: " + fechaimagen);
        rutaimagenprocesada = "/images/ex
tradata/imagenp" + numeroimagenes + "." +
data.extension;
        imagenes.nombreprocesada = rutaim
agenprocesada;
        imagenes.nombre = "/images/extrad
ata/imagen" + numeroimagenes + "." + data
.extension;
      imagenes.fecha = fechaimagen;
      imagenes.errores = data.errores;
      imagenes.numerodientes = data.numer
odientes;
        imagenes.modulo = data.modulo;
        imagenes.paso = data.paso;
        imagenes.diametroexterno = data.d
iametroexterno;

        totalimagenes.push(imagenes);
        var variableimagenes = JSON.strin
gify(totalimagenes);
        fs.writeFile(__dirname + "/public
/images/extradata/imagen" + numeroimagene
s + "." + data.extension, data.cadena64,
'base64', function (err) {
            console.log("Imagen guardada
en:" + __dirname + "/public/images/extrad
ata/imagen" + numeroimagenes + "." + data
.extension);
        });
        fs.writeFile(__dirname + "/public
/images/extradata/imagenp" + numeroimagen
es + "." + data.extension, data.cadena64p
rocesada, 'base64', function (err) {
            console.log("Imagen guardada
en:" + __dirname + "/public/images/extrad
ata/imagenp" + numeroimagenes + "." + dat
a.extension);
        });
        fs.writeFile(__dirname + "/public
/images/imagenes.json", variableimagenes,
'utf8', function (err) {
            console.log("JSON generado co
rrectamente");
        });
        socket.broadcast.emit('nueva_imag
en', imagenes); //Envia a todos los clien
tes
        imagenes = {};
        //Guardar en base de datos
        numeroimagenes += 1;
    });
});
module.exports = {app: app, server: serve
r};
```

ANEXO 4 – CLIENTE WEB.

- **Fichero controlymoni.html (CONTROL Y MONITORIZACIÓN)**

```
<!DOCTYPE html>
<html>
    <!--INFORMACION DE LA PÁGINA -->
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <meta name="description" content="Trabajo Final de Máster en Sistemas Automáticos
y Electrónica Industrial">
        <meta name="keywords" content="Ciberfísicos, IOT, Visión Artifical, Control de Cal
idad">
        <meta name="author" content="Fernando Jácome">
        <title> SECVIA | CONTROL Y MONITORIZACIÓN</title>
        <link rel="stylesheet" href="/stylesheets/style.css">
        <link rel="shortcut icon" href="/images/iot.png">
    </head>
    <body>
        <!-- Encabezado de la pagina WEB -->
        <header>
            <div class="container">
                <div id="branding">
                    <h1> <span class="highlight">  CONTROL Y MONITORIZACIÓN DE UN SISTEMA
FISÍCO DESDE LA RED </span> </h1>
                    <h2> Una introducción interactiva a un Sistema Ciber-físico (CPS)</h2>
                    <h3> SISTEMA DE CLASIFICACIÓN DE ENGRANAJES CON VISIÓN ARTIFICIAL </h3
>
                </div>
                <nav>
                    <ul>
                        <li> <a href="inicio.html" > INICIO</a></li>
                        <li class ="current" > <a href="controlymoni.html" > CONTROL Y MON
ITORIZACIÓN </a></li>
                        <!-- <li > <a href="practicas.html" >DISPOSITIVO 2</a></li> -->
                        <li > <a href="historial.html" > HISTORIAL</a></li>
                        <li> <a href="estadisticas.html" > ESTADISTICAS </a> </li>
                        <li> <a href="docs.html"> DOCUMENTACIÓN</a></li>
                    </ul>
                </nav>
            </div>
        </header>
        <!--SECCION 1 LABORATORIO-->
        <section id="showcase2">
            <div class ="container" style="font-size: 13px">
                <h2 style="font-size-adjust: 25px">SECVIA</h2>
            </div>
        </section>
        <!-- SECCION 2 COMANDOS DE CALIBRACION -->
       <section id ="showcase2" style="background-color:  #268982" style="margin-left: 20p
x" style="margin-right: 20px">
            <table id="tablacomandos"style="width:80%">
                <tr>
                    <td colspan="1">
                        <h2>CALIBRACION DE SENSOR DE POSICIÓN</h2>
                    </td>
                    <td>
                        <h2>ESTADO SENSOR</h2>
                    </td>
                </tr>
                <tr>
                    <td style="background-color:  #268982">
```

```
                        <div class ="container" style="font-weight: bold" style="font-size
-adjust: 10px">
                            <form>
                                <p>  Valor de Umbral </p>
                                <input id="umbral" type="range" name="sliderumbral" min="1
0" max="800" value="20">
                            </form>
                            <label id="umbralvalor" style="font-size: 15px" style="text-al
ign: center" style="color: #000000"> </label>
                            <p id="datocalibrar" style="display:none" > CALIBRADO </p>
                        </div>
                    </td>
                     <td style="background-color:  #268982">
                                <div class="square2" id="valorpresenciamanual"></div>
                            </td>
                </tr>
            </table>
        </section>

        <!-- SECCION 2 COMANDOS DE CONTROL -->
        <section id="controles">
            <div class="container_monitorizacion" id="auto">
                <div class ="comandos">
                    <table id = "tablacomandos" style=" width:80%">
                        <tr>
                            <td COLSPAN=2>
                                <div>
                                    <button id="automatico" > <strong>AUTOMATICO</strong>
</button>
                                </div>
                            </td>
                            <!-- INICIO PARTE MANUAL-->
                           <td COLSPAN=2>
                                <div >
                                    <button id="manual"> <strong> SISTEMA MANUAL</strong>
</button>
                                </div>
                            </td>
                          <!-- FIN PARTE MANUAL-->
                        </tr>
                        <tr>
                            <th>COMANDOS</th>
                            <th>ESTADO</th>
                        <!--    <th>COMANDOS</th> -->
                            <th>OBSERVACION</th>
                        </tr>
                        <tr>
                            <td>
                                <div>
                                    <button id="start"> <strong>ARRANQUE DEL SISTEMA</stro
ng> </button>
                                </div>
                            </td>
                            <td>
                                <div class="square2" id="startcolor"></div>
                            </td>
                            <!-- INICIO PARTE MANUAL-->
                           <td>
                                <button id="motoradelante" value="0"> <strong>MOTOR ADELAN
TE</strong> </button>
                            </td>
                         <!-- FIN PARTE MANUAL-->
                            <td>
                                <p id="textstart" style="display:none"> SISTEMA EN FUNCION
AMIENTO </p>
```

```
                                <p id="motoradelantelabel" style="display:none"> MOTOR HAC
IA ADELANTE </p>

                            </td>
                        </tr>
                        <tr>
                            <td>
                                <button id ="stop"> <strong>PARO DEL SISTEMA</strong> </bu
tton>
                            </td>
                            <td>
                                <div class="square" id="stopcolor"></div>
                            </td>
                            <!-- INICIO PARTE MANUAL-->
                           <td>
                                <button id="motoratras" > <strong>MOTOR ATRAS</strong> </b
utton>
                            </td>
                          <!-- FIN PARTE MANUAL-->
                            <td>
                                <p id="textstop" style="display:none"> SISTEMA EN STANBY!!
!!!! </p>
                                <p id="motoratraslabel" style="display:none"> MOTOR HACIA
ATRAS </p>
                            </td>
                        </tr>
                        <tr>
                            <td>
                                <h5  id ="botonemergencia">EMERGENCIA </h5>
                            </td>
                            <td>
                                <input id="emergencia" type="checkbox" value="0"  > <div c
lass="square2" id="emergenciacolor"></div>
                            </td>
                            <!-- INICIO PARTE MANUAL-->
                           <td>
                                <button id="capturarengrane"> <strong> PROCESAR ENGRANE</s
trong> </button>
                            </td>
                         <!-- FIN PARTE MANUAL-->
                            <td>
                                <p id="textemergencia" style="display:none"> RETIRAR EMERG
ENCIA!!!!!! </p>
                                <p id="textcapturarengrane" style="display:none"> ENGRANE
CAPTURADO Y PROCESADO </p>

                            </td>
                        </tr>
                        <tr>
                            <td COLSPAN=3> <form>
                                    VELOCIDAD DE MOTOR
                                    <input id="motor" type="range" name="slidermotor" min=
"0" max="100" value="0">
                                </form>
                            </td>
                            <td>
                                <p> VELOCIDAD DEL MOTOR </p>
                                <label id="motorvalor" style="font-size:15px" style="text-
align: center" style="color: #000000"> </label>
                            </td>
                        </tr>
                        <tr>
                            <td>
                                <strong>SENSOR DE PRESENCIA</strong>
                            </td>
                            <td>
```

```
                                <div class="square2" id="valorpresencia"></div>
                            </td>
                            <!-- INICIO PARTE MANUAL-->
                           <td>
                                <button id="clasificacionbueno"> <strong>CLASIFICADOR BUEN
O</strong> </button>
                            </td>
                         <!-- FIN PARTE MANUAL-->
                            <td>
                                <p id="textconobjeto" style="display:none"> PRESENCIA DE E
NGRANAJE </p>
                                <p id="textsinobjeto" style="display:none"> VACIO!!!! </p>
                                <p id="engranebueno" style="display:none"> ENGRANE BUENO!!
!! </p>

                            </td>
                        </tr>
                        <tr>
                            <td>
                                <strong>CLASIFICADOR</strong>
                            </td>
                            <td>
                                <div class="square" id="estadoclasifi"></div>
                            </td>
                            <!-- INICIO PARTE MANUAL-->
                            <td>
                                <button id="clasificacionmalo"> <strong>CLASIFICADOR MALO<
/strong> </button>
                            </td>
                            <!-- FIN PARTE MANUAL-->
                            <td>
                                <p id="textobjetobueno" style="display:none"> ENGRANE BUEN
O </p>
                                <p id="textobjetomalo" style="display:none"> ENGRANE MALO
</p>
                                <p id="engranemalo" style="display:none"> ENGRANE MALO!!!!
</p>
                            </td>
                        </tr>
                    </table>
                </div>
                <div class="comandos">
                    <strong>IMAGEN PROCESADA</strong><br>
                    <img src="/images/EPSEVG.jpg"  style="width:40%" id="imagen_actual" al
t="Salida de imagen">
                    <label id="detalleengrane"> </label>
                    <button onclick="window.location.href = 'historial.html'"> VER DETALLE
S </button>
                </div>
            </div>
        </section>



        <!--SECCION 4 ENLACES UNIVERSIDAD-->
        <section id="boxes">
            <div class ="container_boxes">
                <table id = "tablaenlaces" style=" width:95%">
                    <tr>
                        <th>
                            <p> <a href="https://www.upc.edu/ca"> UPC </a> </p>
                        </th>
                        <th>
                            <a href="https://www.epsevg.upc.edu/ca">EPSEVG </a>
                        </th>
```

```
                    </tr>
                    <tr>
                        <td>
                            <p>
                                <a href="https://www.upc.edu/ca"> <br>
                                    <img src="/images/Logo_UPC.png" style="width:25%" alt=
"UPC"> </a>
                            </p>
                        </td>
                        <td>
                            <p>
                                <a href="https://www.epsevg.upc.edu/ca">
                                    <img src="/images/Logo_epsevg.png" style="width:25%" a
lt="EPSEVG "> </a>
                            </p>
                        </td>

                    </tr>
                    <tr>
                        <td>
                            <p> Universidad Politècnica de Cataluña </p>
                        </td>
                        <td>
                            <p> Escuela Politécnica Superior de Ingenieria de Vilanova y G
eltrú</p>
                        </td>

                    </tr>
                </table>
            </div>
        </section>
        <!--PIE DE PAGINA -->
        <footer style="font-size-adjust: 10px">
            <p> Escuela Politécnica Superior de Ingeniería de Vilanova i la Geltrú. EPSEVG
Av. Víctor Balaguer &copy; 08800 Vilanova i la Geltrú - Tel.: 93 896 77 01 / 93 896 77 02
</p>
        </footer>
        <script src="/socket.io/socket.io.js"></script>
        <script src="/javascripts/main.js"></script>
    </body>
</html>
```

- Entorno de programación CONTROL Y MONITORIZACIÓN (main.js)

```
/*
 * To change this license header, choose
License Headers in Project Properties.
 * To change this template file, choose T
ools | Templates
 * and open the template in the editor.
 */

var socket = io(); // CREA UNA VARIABLE P
ARA ENVIAR AL CLIENTE DEPEDNIEDO DE LA VA
RIABLE EN HTML
var valormotor = document.getElementById(
"motor");
var labelmotor = document.getElementById(
"motorvalor");
var valoremergencia = document.getElement
ById("emergencia");
var valorstart = document.getElementById(
"start");
var observacionstart = document.getElemen
tById("obserstart");
var valorstop = document.getElementById("
stop");
var estadoclasificador = document.getElem
entById("estadoclasifi");
var estadopresencia = document.getElement
ById("valorpresencia");
```

```
var imagen1 = document.getElementById("im
agen_actual");
var reporteerrores = document.getElementB
yId("reporteerrores");
var fecha = document.getElementById('fech
a');
var colorstart = document.getElementById(
'startcolor');
var colorstop = document.getElementById('
stopcolor');
var coloremergencia = document.getElement
ById('emergenciacolor');
var botemergencia = document.getElementBy
Id("botonemergencia");
var textemerg = document.getElementById("
textemergencia");
var texobjbueno = document.getElementById
("textobjetobueno");
var texobjmalo = document.getElementById(
"textobjetomalo");
var textparo = document.getElementById("t
extstop");
var textrun = document.getElementById("te
xtstart");
var textcongear = document.getElementById
("textconobjeto");
var textsingear = document.getElementById
("textsinobjeto");
var infoengranaje = document.getElementBy
Id("detalleengrane");
var sistemaautomatico = document.getEleme
ntById("automatico");

//INICIO PARTE MANUAL
// AÑADIR PARTE MANUAL
var estadopresenciamanual = document.getE
lementById("valorpresenciamanual");
var sistemamanual = document.getElementBy
Id("manual");
var motoradelante = document.getElementBy
Id("motoradelante");
var motoradelantelabel = document.getElem
entById("motoradelantelabel");
var motoratras = document.getElementById(
"motoratras");
var motoratraslabel = document.getElement
ById("motoratraslabel");
var capturarengrane = document.getElement
ById("capturarengrane");
var textcapturarengrane = document.getEle
mentById("textcapturarengrane");
var clasificacionbueno = document.getElem
entById("clasificacionbueno");
var engranebueno = document.getElementByI
d("engranebueno");
var clasificacionmalo = document.getEleme
ntById("clasificacionmalo");
var engranemalo = document.getElementById
("engranemalo");
var umbral = document.getElementById("umb
ral");
var umbralvalor = document.getElementById
("umbralvalor");
var datocalibrar = document.getElementByI
d("datocalibrar");
//FIN PARTE MANUAL


var mensaje;

var arreglodispositivo = []; // inicia en
forma de vector

sistemaautomatico.addEventListener("click
", function () { // AÑADE UNA LISTA EL MO
TOR DESPUE DE DAR UN CLICK
    var dispositivo = {};
    dispositivo.dispositivo = 1;
    dispositivo.modofunc = true;
    dispositivo.estado = 0;
     valoremergencia.disabled = false;
    valorstart.disabled = false;
    valorstop.disabled = false;
    texobjbueno.style.display = "none";
    texobjmalo.style.display = "none";
    textcongear.style.display = "none";
    textemerg.style.display = "none";
    textrun.style.display = "none";
    textsingear.style.display = "none";
    textparo.style.display = "none";

    //INICIO PARTE manual
       motoradelante.disabled = true;
    motoratras.disabled = true;
    clasificacionbueno.disabled = true;
    capturarengrane.disabled = true;
    clasificacionmalo.disabled = true;
    motoradelantelabel.style.display = "n
one";
    motoratraslabel.style.display = "none
";
    textcapturarengrane.style.display = "
none";
    engranebueno.style.display = "none";
    engranemalo.style.display = "none";
 // fin parte manual



    arreglodispositivo.push(dispositivo);
    console.log(arreglodispositivo);
socket.emit('datos', arreglodispositivo);
    arreglodispositivo = [];
});

//INICIO PARTE MANUAL
sistemamanual.addEventListener("click", f
unction () { // AÑADE UNA LISTA EL MOTOR
DESPUE DE DAR UN CLICK
    var dispositivo = {};
    dispositivo.dispositivo = 1;
    dispositivo.modofunc = false;
    dispositivo.direcclasificacion = 0;
    dispositivo.sentidomotor=0;
    motoradelante.disabled = false;
    motoratras.disabled = false;
    capturarengrane.disabled = false;
    clasificacionbueno.disabled = false;
    clasificacionmalo.disabled = false;
    valoremergencia.disabled = true;
    valorstart.disabled = true;
    valorstop.disabled = true;
    engranebueno.style.display = "none";
```

```
      motoradelantelabel.style.display =
"none";
       motoratraslabel.style.display = "n
one";
    texobjbueno.style.display = "none";
    texobjmalo.style.display = "none";
    textcongear.style.display = "none";
    textemerg.style.display = "none";
    textrun.style.display = "none";
    textsingear.style.display = "none";
    textparo.style.display = "none";
    arreglodispositivo.push(dispositivo);
    console.log(arreglodispositivo);
    socket.emit('datos', arreglodispositi
vo);
    arreglodispositivo = [];
});
//INICIO PARTE MANUAL


valormotor.addEventListener("click", func
tion () { // AÑADE UNA LISTA EL MOTOR DES
PUE DE DAR UN CLICK
    var dispositivo = {};

        dispositivo.motor = Number(valorm
otor.value);
    labelmotor.innerHTML = valormotor.val
ue;
    arreglodispositivo.push(dispositivo);
    console.log(arreglodispositivo);
    socket.emit('datos', arreglodispositi
vo);
    arreglodispositivo = [];
});
valoremergencia.addEventListener("click",
function () {
    var dispositivo = {};
    if (valoremergencia.checked) {
        dispositivo.dispositivo = 1;
        dispositivo.modofunc = true;
        dispositivo.estado = -1;
        valorstart.disabled = true;
        valorstop.disabled = true;
        textemerg.style.display = "block"
;
        textrun.style.display = "none";
        textparo.style.display = "none";
        coloremergencia.style.backgroundC
olor = 'Red';
        colorstart.style.backgroundColor
= "#f4f4f4";//gris oscuro
        colorstop.style.backgroundColor =
"#35424a";
    } else {
        dispositivo.dispositivo = 1;
        dispositivo.modofunc = true;
        dispositivo.estado = 0;
        console.log("StopTomate");
        colorstop.style.backgroundColor =
'Orange';
        colorstart.style.backgroundColor
= "#f4f4f4";//gris oscuro
        coloremergencia.style.backgroundC
olor = "#f4f4f4";//gris oscuro
        valorstart.disabled = false;
        valorstop.disabled = true;
        textemerg.style.display = "none";
        textparo.style.display = "block";

    }
    arreglodispositivo.push(dispositivo);
    console.log(arreglodispositivo);
    socket.emit('datos', arreglodispositi
vo);
    arreglodispositivo = [];
});


botemergencia.addEventListener("click", f
unction () {
    var dispositivo = {};
    dispositivo.dispositivo = 1;
    dispositivo.modofunc = true;
    dispositivo.estado = -1;
    valoremergencia.checked = true;
    valorstart.disabled = true;
    valorstop.disabled = true;
    coloremergencia.style.backgroundColor
= 'Red';
    colorstart.style.backgroundColor = "#
f4f4f4";//gris oscuro
    colorstop.style.backgroundColor = "#3
5424a";
    textemerg.style.display = "block";
    textrun.style.display = "none";
    textparo.style.display = "none";

    arreglodispositivo.push(dispositivo);
    console.log(arreglodispositivo);
    socket.emit('datos', arreglodispositi
vo);
    arreglodispositivo = [];
});
valorstart.addEventListener("click", func
tion () {
    var dispositivo = {};
    dispositivo.dispositivo = 1;
    dispositivo.modofunc = true;
    dispositivo.estado = 1;
    valorstart.disabled = true;
    valorstop.disabled = false;
    colorstart.style.backgroundColor = 'G
reen';
    colorstop.style.backgroundColor = "#3
5424a";
    coloremergencia.style.backgroundColor
= "#f4f4f4";//gris oscuro;
    textrun.style.display = "block";
    textparo.style.display = "none";
    arreglodispositivo.push(dispositivo);
    console.log(arreglodispositivo);
    socket.emit('datos', arreglodispositi
vo);
    arreglodispositivo = [];
});

valorstop.addEventListener("click", funct
ion () {
    var dispositivo = {};
    dispositivo.dispositivo = 1;
    dispositivo.modofunc = true;
```

```
    dispositivo.estado = 0;
    valorstop.disabled = true;
    valorstart.disabled = false;
    console.log("StopTomate");
    colorstop.style.backgroundColor = 'Or
ange';
    colorstart.style.backgroundColor = "#
f4f4f4";//gris oscuro
    coloremergencia.style.backgroundColor
= "#f4f4f4";//gris oscuro
    textparo.style.display = "block";
    textrun.style.display = "none";
    arreglodispositivo.push(dispositivo);
    console.log(arreglodispositivo);
    socket.emit('datos', arreglodispositi
vo);
    arreglodispositivo = [];
});
//INICIO PARTE MANUAL

//AÑADIR LOS COMANDOS PARA LOS DEMAS
umbral.addEventListener("click", function
() { // AÑADE UNA LISTA EL MOTOR DESPUE D
E DAR UN CLICK
    var dispositivo = {};
    dispositivo.dispositivo = 1;
    dispositivo.modofunc = false;
    dispositivo.capturarengrane = false;
    dispositivo.umbral = Number(umbral.va
lue);
    umbralvalor.innerHTML = umbral.value;
    if (umbral.value >= 200 & umbral.valu
e <= 300) {

        datocalibrar.style.display = "blo
ck";
    } else {
        datocalibrar.style.display = "non
e";
    }
    arreglodispositivo.push(dispositivo);
    console.log(arreglodispositivo);
    socket.emit('datos', arreglodispositi
vo);
    arreglodispositivo = [];
});
motoradelante.addEventListener("click", f
unction () {
    var dispositivo = {};
    dispositivo.dispositivo = 1;
    dispositivo.modofunc = false;
     dispositivo.direcclasificacion = 0;
     dispositivo.sentidomotor = 1;
     dispositivo.capturarengrane = false;
        motoradelantelabel.style.display
= "block";
        motoratraslabel.style.display = "
none";
    arreglodispositivo.push(dispositivo);
    console.log(arreglodispositivo);
    socket.emit('datos', arreglodispositi
vo);
    arreglodispositivo = [];
});
motoratras.addEventListener("click", func
tion () {
    var dispositivo = {};
    dispositivo.dispositivo = 1;
    dispositivo.modofunc = false;
    dispositivo.direcclasificacion = 0;
    dispositivo.sentidomotor = -1;
     dispositivo.capturarengrane = false;
    console.log("MOTOR ATRAS");
    motoradelantelabel.style.display = "n
one";
    motoratraslabel.style.display = "bloc
k";
    arreglodispositivo.push(dispositivo);
    console.log(arreglodispositivo);
    socket.emit('datos', arreglodispositi
vo);
    arreglodispositivo = [];

});

capturarengrane.addEventListener("click",
function () {
    var dispositivo = {};
    dispositivo.dispositivo = 1;
    dispositivo.modofunc = false;
    dispositivo.sentidomotor = 0;
    dispositivo.direcclasificacion = 0;
    dispositivo.capturarengrane = true;
    console.log("imagen capturada");
    textcapturarengrane.style.display = "
block";
    arreglodispositivo.push(dispositivo);
    console.log(arreglodispositivo);
    socket.emit('datos', arreglodispositi
vo);
    arreglodispositivo = [];
});

clasificacionbueno.addEventListener("clic
k", function () {
    var dispositivo = {};
    dispositivo.dispositivo = 1;
    dispositivo.modofunc = false;
    dispositivo.sentidomotor = 0;
    dispositivo.direcclasificacion = 1;
    dispositivo.capturarengrane = false;
    console.log(" engrane bueno");
    engranebueno.style.display = "block";
    engranemalo.style.display = "none";
    arreglodispositivo.push(dispositivo);
    console.log(arreglodispositivo);
    socket.emit('datos', arreglodispositi
vo);
    arreglodispositivo = [];
});
clasificacionmalo.addEventListener("click
", function () {
    var dispositivo = {};
    dispositivo.dispositivo = 1;
    dispositivo.modofunc = false;
    dispositivo.sentidomotor = 0;
    dispositivo.direcclasificacion = -1;
    dispositivo.capturarengrane = false;
    console.log(" engrane malo");
    engranemalo.style.display = "block";
    engranebueno.style.display = "none";
    arreglodispositivo.push(dispositivo);
```

```
    console.log(arreglodispositivo);
    socket.emit('datos', arreglodispositi
vo);
    arreglodispositivo = [];
});
//FIN PARTE MANUAL
//
console.log("Inicio del Script");
socket.on('datos', function (data) { //ge
t button status from client
    var sizedatos = data.length;
    console.log(data);
    console.log("Tamaño del dato en Datos
: " + sizedatos);
    for (i = 0; i < sizedatos; i++) {
        if (data[i].motor != null) {
valormotor.value = Number(data[i].motor);
      labelmotor = Number(data[i].motor);
        }
        if (data[i].presencia != null) {
        if (data[i].presencia == true) {
           console.log("Presenciaazul");
    estadopresencia.style.backgroundColor
= 'Blue';
    estadopresenciamanual.style.backgroun
dColor = 'Blue';
     textcongear.style.display = "block";
            } else {
     console.log("presenciaGris");
    estadopresencia.style.backgroundColor
= "#f4f4f4";
  estadopresenciamanual.style.backgroundC
olor = "#268982";
    textsingear.style.display = "block";
            }
        }
      if (data[i].clasificador != null) {
      if (data[i].clasificador == true) {
        console.log("clasiAZUL");
          estadoclasificador.style.backgr
oundColor = "#6600cc";
     texobjbueno.style.display = "block";
     texobjmalo.style.display = "none";
      } else {
        estadoclasificador.style.backgrou
ndColor = "#f4f4f4";
        console.log("clasiGRIS");
      texobjbueno.style.display = "none";
      texobjmalo.style.display = "block";
            }
        }
        if (data[i].estado != null) {
            if (data[i].estado == -1) {
                console.log("emerROJO");
   coloremergencia.style.backgroundColor
= 'Red';
    colorstart.style.backgroundColor = "#
f4f4f4";//gris oscuro
colorstop.style.backgroundColor = "#35424
a";
  valorstart.disabled = true;
   valorstop.disabled = true;
                valoremergencia.checked =
true;
            } else if (data[i].estado ==
1) {
    console.log("startVerde");
   colorstart.style.backgroundColor = 'Gr
een';
   colorstop.style.backgroundColor = "#35
424a";
   coloremergencia.style.backgroundColor
= "#f4f4f4";
   valorstart.disabled = true;
   valorstop.disabled = false;
   valoremergencia.checked = false;
      } else if (data[i].estado == 0) {
       console.log("StopTomate");
       colorstop.style.backgroundColor =
'Orange';
    colorstart.style.backgroundColor = "#
f4f4f4";//gris oscuro
  coloremergencia.style.backgroundColor =
"#f4f4f4";//gris oscuro
  valorstart.disabled = false;
   valorstop.disabled = true;
      valoremergencia.checked = false;
            }
        }
        //INICIO PARTE MANUAL
       if (data[i].umbral != null) {
   umbral.value = Number(data[i].umbral);
    umbralvalor = Number(data[i].umbral);
      }
     if (data[i].modofunc != null) {
         if (data[i].modofunc == true) {
          sistemamanual.disabled = false;
           motoradelante.disabled = true;
             motoratras.disabled = true;
         capturarengrane.disabled = true;
      clasificacionbueno.disabled = true;
       clasificacionmalo.disabled = true;
        motoradelantelabel.style.display
= "none";
  motoratraslabel.style.display = "none";
    engranebueno.style.display = "none";
      engranemalo.style.display = "none";
            } else {
           sistemamanual.disabled = true;
             valorstart.disabled = true;
          motoradelante.disabled = false;
           motoratras.disabled = false;
       capturarengrane.disabled = false;
    clasificacionbueno.disabled = false;
     clasificacionmalo.disabled = false;
     textcapturarengrane.style.display =
"none";
     engranebueno.style.display = "none";
motoradelantelabel.style.display = "none"
;
  motoratraslabel.style.display = "none";
            }
        }
  if (data[i].sentidomotor != null) {
    if (data[i].sentidomotor == -1) {
    console.log("MOTOR ATRAS");
motoradelantelabel.style.display ="none";
 motoratraslabel.style.display = "block";
  } else if (data[i].sentidomotor == 1) {
   console.log("MOTOR ADELANTE");
```

```
motoradelantelabel.style.display ="block"
;
motoratraslabel.style.display = "none";
} else if (data[i].sentidomotor == 0) {
}
}
if (data[i].direcclasificacion != null) {
  if (data[i].direcclasificacion == -1) {
    console.log(" engrane malo");
   engranemalo.style.display = "block";
   engranebueno.style.display = "none";
    } else if (data[i].direcclasificacion
== 1) {
   console.log(" engrane bueno");
    engranebueno.style.display = "block";
      engranemalo.style.display = "none";
    } else if (data[i].direcclasificacion
== 0) {
            }
        }
        //FIN PARTE MANUAL
    }
});
socket.on('nueva_imagen', function (data)
{
    imagen1.src = data.nombreprocesada;
    infoengranaje.innerHTML = "Numero de
dientes: " + data.numerodientes +
            "<br> " + "Modulo: " + data.m
odulo.toFixed(2) + "<br> " + "Paso: " + d
ata.paso.toFixed(2) + "<br> " + "Diametro
exterior: " + data.diametroexterno.toFixe
d(2) + "mm";
    if (data.errores === " ") {
        reporteerrores.innerHTML = "Pieza
sin errores!";
  texobjbueno.innerHTML = "Objeto bueno";
    } else {
 reporteerrores.innerHTML = data.errores;
   texobjbueno.innerHTML = "Objeto descar
tado";
    }
    fecha.innerHTML = "Fecha: " + data.fe
cha;

});

socket.on('nuevousuario', function (data)
{ //get button status from client
    console.log("nuevo usuario");
    console.log(data);
    var sizedatos = data.length;
    console.log(typeof data);
    console.log("Tamaño del dato" + sized
atos);
    for (i = 0; i < sizedatos; i++) {
        if (data[i].motor != null) {
            valormotor.value = Number(dat
a[i].motor);
            labelmotor = Number(data[i].m
otor);
        }
        if (data[i].presencia != null) {
            if (data[i].presencia == true
) {
console.log("Presenciaazul");
  estadopresencia.style.backgroundColor =
'Blue';
   textcongear.style.display = "block";
            } else {
    console.log("presenciaGris");
     estadopresencia.style.backgroundColo
r = "#35424a";
     textsingear.style.display = "block";
            }
        }
    if (data[i].clasificador != null) {
            if (data[i].clasificador == t
rue) {
    console.log("clasiAZUL");
   estadoclasificador.style.backgroundCol
or ="#6600cc";// ojo cambiar los colortes
    texobjbueno.style.display = "block";
   texobjmalo.style.display = "none";
            } else {
 estadoclasificador.style.backgroundColor
= "#f4f4f4";
    console.log("clasiGRIS");
    texobjbueno.style.display = "none";
   texobjmalo.style.display = "block";
            }
        }
   if (data[i].estado != null) {
            if (data[i].estado == -1) {
    console.log("emerROJO");
   coloremergencia.style.backgroundColor
= 'Red';
  colorstart.style.backgroundColor = "#35
424a";//gris oscuro
 colorstop.style.backgroundColor = "#f4f4
f4";
   valorstart.disabled = true;
  valorstop.disabled = true;
  valoremergencia.checked = true;
    } else if (data[i].estado == 1) {
    console.log("startVerde");
    colorstart.style.backgroundColor = 'G
reen';
      colorstop.style.backgroundColor = "
#f4f4f4";
  coloremergencia.style.backgroundColor =
"#35424a";
    valorstart.disabled = true;
     valorstop.disabled = false;
   valoremergencia.checked = false;
    } else if (data[i].estado == 0) {
    console.log("StopTomate");
colorstop.style.backgroundColor = 'Orange
';
  colorstart.style.backgroundColor = "#35
424a";//gris oscuro
  coloremergencia.style.backgroundColor =
"#35424a";//gris oscuro
   valorstart.disabled = false;
 valorstop.disabled = true;
   valoremergencia.checked = false;
            }
        }
        //INICIO PARTE MANUAL
  if (data[i].umbral != null) {
   umbral.value = Number(data[i].umbral);
    umbralvalor = Number(data[i].umbral);
```

```
        }
    if (data[i].modofunc != null) {
     if (data[i].modofunc == true) {
     sistemamanual.disabled = false;
      motoradelante.disabled = true;
       motoratras.disabled = true;
  capturarengrane.disabled = true;
     clasificacionbueno.disabled = true;
       clasificacionmalo.disabled = true;
 motoradelantelabel.style.display = "none
";
 motoratraslabel.style.display = "none";
    engranebueno.style.display = "none";
     engranemalo.style.display = "none";
            } else {
     sistemamanual.disabled = true;
     valorstart.disabled = true;
      motoradelante.disabled = false;
      motoratras.disabled = false;
     capturarengrane.disabled = false;
     clasificacionbueno.disabled = false;
     clasificacionmalo.disabled = false;
textcapturarengrane.style.display = "none
";
engranebueno.style.display = "none";
motoradelantelabel.style.display = "none"
;
  motoratraslabel.style.display = "none";
            }
         }
if (data[i].sentidomotor != null) {
 if (data[i].sentidomotor == -1) {
   console.log("MOTOR ATRAS");
motoradelantelabel.style.display ="none";
motoratraslabel.style.display = "block";
} else if (data[i].sentidomotor == 1) {
console.log("MOTOR ADELANTE");
motoradelantelabel.style.display = "block
";
   motoratraslabel.style.display = "none"
;
  } else if (data[i].sentidomotor == 0) {
     }
  }
if (data[i].direcclasificacion != null) {
if (data[i].direcclasificacion == -1) {
    console.log(" engrane malo");
     engranemalo.style.display = "block";
   engranebueno.style.display = "none";
      } else if (data[i].direcclasificaci
on == 1) {
    console.log(" engrane bueno");
    engranebueno.style.display = "block";
    engranemalo.style.display = "none";
       } else if (data[i].direcclasifica
cion == 0) {
     }
  }
       //FIN PARTE MANUAL
 }

});
socket.on('lastimagen', function (data) {
    imagen1.src = data;
});
function cambioPestaña(evento, modo) {
    var i, tabcontent, tablinks;
    // Get all elements with class="tabco
ntent" and hide them
    tabcontent = document.getElementsByCl
assName("contenedor");
    for (i = 0; i < tabcontent.length; i+
+) {
    tabcontent[i].style.display = "none";
    }
    // Get all elements with class="tabli
nks" and remove the class "active"
    tablinks = document.getElementsByClas
sName("tablinks");
    for (i = 0; i < tablinks.length; i++)
{
        tablinks[i].className = tablinks[
i].className.replace(" active", "");
    }

    // Show the current tab, and add an "
active" class to the button that opened t
he tab
    document.getElementById(modo).style.d
isplay = "block";
    evento.currentTarget.className += " a
ctive";
}
```

ANEXO 5 – CLIENTE FÍSICO (PROTOTIPO SECVIA).

```
/*
 * File:   main.cpp
 * Author: Fernando Jácome
 *
 * Created on July 9, 2018, 12:55 PM
 */


#include "sio_client.h"                 /
/Librería de Sockets io
#include "base64.h"                     /
/Libreria para transformar strings a base
64
#include <functional>                   /
/Librerías estándar
#include <iostream>
#include <thread>                       /
/Librerías de Multriprocesamiento
#include <mutex>
#include <condition_variable>
#include <string>                       /
/Librerías de cadena de caracteres
#include <sstream>
#include <fstream>                      /
/Libreria para abrir archivos
#include <vector>                       /
/Manejo de vectores
#include <chrono>                       /
/Libreria para fecha y año
#include <cstdlib>
#include <stdio.h>
#include <errno.h>
#include <opencv2/highgui.hpp>
#include "ArduinoJson.h"                /
/Libreria para el Manejo de Cadenas Json
en C++


#ifdef __arm__              //Compila solo
si se encuentra dentro del Raspberry Pi(p
rocesador arm)

#include <wiringPi.h>
#include <wiringSerial.h>
#endif
#define HIGHLIGHT(__O__) std::cout<<"\e[1
;31m"<<__O__<<"\e[0m"<<std::endl   //Colo
rear las funciones en el terminal
#define EM(__O__) std::cout<<"\e[1;30;1m"
<<__O__<<"\e[0m"<<std::endl
int fd;
int fd2;
#include "Vision.h"                    //L
ibreria y funciones de vision para engran
es

using namespace sio; //Workspace de Socke
ts.io
using namespace std; //Workspace de std
using namespace cv; //Workspace de opEnCV

std::mutex _lock; //Workspace de multithr
eading
std::condition_variable_any _cond;
bool connect_finish = false;
//Variables que se pueden modificar a tra
vés de los argumentos de C
string servidorIP = "127.0.0.1";
string puerto = "3000";
string puertoserial = "/dev/ttyUSB0";
int idcamara = 0; // La primera camara qu
e se encuentra en /dev/video0

vector<uchar> buf; //Buffer para guardar
los datos de la imagen
std::string encoded, encodedprocesada, s,
tipo;
std::time_t tt;
bool datos = false;
bool modfun;
//Variables para el envio de datos
const size_t bufferSize = JSON_OBJECT_SIZ
E(11);
DynamicJsonBuffer Bufferentrada(bufferSiz
e);
DynamicJsonBuffer Buffersalida(bufferSize
);
JsonObject& datoenvioarduino = Buffersali
da.createObject();
JsonObject& datoenvioarduinomanual = Buff
ersalida.createObject();
JsonObject& datoenvioarduinomotor = Buffe
rsalida.createObject();
std::string cadenapararduino;
std::string cadenapararduinovid;
std::string cadenapararduinomanual;
std::string cadenapararduinomotor;
bool capturarengrane;

const size_t bufferSizeeq2 = JSON_ARRAY_S
IZE(3) + JSON_OBJECT_SIZE(2);
DynamicJsonBuffer Bufferentradaeq2(buffer
Sizeeq2);
DynamicJsonBuffer Buffersalidaeq2(bufferS
izeeq2);
JsonObject& datoenvioarduinoeq2 = Buffers
alidaeq2.createObject();

std::string cadenapararduinoeq2;

bool iniciocadena = false, fincadena = fa
lse;
bool iniciocadenaeq2 = false, fincadenaeq
2 = false;

/*
 *  Clase para conectar el listener, para
aguardar por eventos
 */
class connection_listener {
    sio::client &handler;

public:
```

```
    connection_listener(sio::client& h) :
    handler(h) {
    }

    void on_connected() {
        _lock.lock();
        _cond.notify_all();
        connect_finish = true;
        _lock.unlock();
        std::cout << "Servidor conectado
correctamente" << std::endl;
    }

    void on_close(client::close_reason co
nst& reason) {
        std::cout << "sio closed " << std
::endl;
        exit(0);
    }

    void on_fail() {
        std::cout << "TFM- SISTEMAS CYBER
FISICOSio failed " << std::endl;
        exit(0);
    }
};

socket::ptr current_socket; //Api de sock
et, el Socket que maneja todos los evento
s de Socket.io

void eventos() {
    current_socket->on("datos", sio::sock
et::event_listener_aux([&](string
            const& name, message::ptr con
st& data, bool isAck, message::list & ack
_resp) {
        _lock.lock(); //Bloquear la ejecu
cion hasta realizar una opcion
std::vector<std::shared_ptr < message>> v
ectoringresojson = data -> get_vector();
//Creo vector para el ingreso de datos

for (int i = 0; i < vectoringresojson.siz
e(); i++) {
            std::map<std::string, message
::ptr> mapa = vectoringresojson.at(i)->ge
t_map();
std::cout << "Vector ingreso json a mapad
atos" << std::endl;
 if (mapa.find("motor") != mapa.end()) {
std::cout << "Motor: " << mapa.find("moto
r")->second->get_int() << std::endl;
datoenvioarduinomotor["motor"] = mapa.fin
d("motor")->second->get_int();

std::cout << "Cadena para Equipo 1 motor:
";
cadenapararduinomotor = "";

datoenvioarduinomotor.printTo(cadenaparar
duinomotor);
// cadenapararduino += std::string("\0");
//Fin de cadena
const char *cstr = cadenapararduinomotor.
c_str();
std::cout << cstr << std::endl;
#ifdef __arm__
  serialPuts(fd, cstr);
#endif
std::cout <<"cadenaresetea" << std::endl;

cadenapararduinomotor = "";
std::cout << "cadenaresetea" << cadenapar
arduinomotor << std::endl;
  }
if (mapa.find("dispositivo") != mapa.end(
)) {
if (mapa.find("dispositivo") != mapa.end(
)) {
std::cout << "dispositivo : " << mapa.fin
d("dispositivo")->second->get_int() << st
d::endl;
datoenvioarduino["dispositivo"] = mapa.fi
nd("dispositivo")->second->get_int();
datoenvioarduinomanual["dispositivo"] = m
apa.find("dispositivo")->second->get_int(
);
}
if (mapa.find("modofunc") != mapa.end())
{
std::cout << "modofunc: " << mapa.find("m
odofunc")->second->get_bool() << std::end
l;
 modfun = mapa.find("modofunc")->second->
get_bool();
}
if (modfun == true) {
  datoenvioarduino["modofunc"] = mapa.fin
d("modofunc")->second->get_bool();
if (mapa.find("motor") != mapa.end()) {
std::cout << "Motor: " << mapa.find("moto
r")->second->get_int() << std::endl;
 datoenvioarduino["motor"] = mapa.find("m
otor")->second->get_int();
                    }
if (mapa.find("estado") != mapa.end()) {
std::cout << "estado: " << mapa.find("est
ado")->second->get_int() << std::endl;
signed int datoestado = mapa.find("estado
")->second->get_int();
datoenvioarduino["estado"] = datoestado;
                }
std::cout << "Cadena para Equipo 1 automa
tico: ";
cadenapararduino = "";
datoenvioarduino.printTo(cadenapararduino
);
 // cadenapararduino += std::string("\0")
; //Fin de cadena
const char *cstr = cadenapararduino.c_str
();
std::cout << cstr << std::endl;
#ifdef __arm__
serialPuts(fd, cstr);
#endif
std::cout << "cadenaresetea" << std::endl
;
cadenapararduino = "";
```

```
std::cout << "cadenaresetea" << cadenapar
arduino << std::endl;
} else {
datoenvioarduinomanual["modofunc"] = mapa
.find("modofunc")->second->get_bool();
if (mapa.find("sentidomotor") != mapa.end
()) {
std::cout << "sentidomotor: " << mapa.fin
d("sentidomotor")->second->get_int() << s
td::endl;
datoenvioarduinomanual["sentidomotor"] =
mapa.find("sentidomotor")->second->get_in
t();
}

if (mapa.find("direcclasificacion") != ma
pa.end()) {
std::cout << "direcclasificacion: " << ma
pa.find("direcclasificacion")->second->ge
t_int() << std::endl;
datoenvioarduinomanual["direcclasificacio
n"] = mapa.find("direcclasificacion")->se
cond->get_int();
                    }
if (mapa.find("umbral") != mapa.end()) {
std::cout << "Umbral IF: " << mapa.find("
umbral")->second->get_int() << std::endl;
datoenvioarduinomanual["umbral"] = mapa.f
ind("umbral")->second->get_int();
                    }
                    if (mapa.find("captur
arengrane") != mapa.end()) {
                        std::cout << "cap
turarengrane: " << mapa.find("capturareng
rane")->second->get_bool() << std::endl;
                                datoenvio
arduinomanual["capturarengrane"] = mapa.f
ind("capturarengrane")->second->get_bool(
);
                                bool capt
urarengrane = mapa.find("capturarengrane"
)->second->get_bool();
                                datoenvio
arduinomanual.printTo(cadenapararduinoman
ual);
// cadenapararduino += std::string("\0");
//Fin de cadena
    const char *cstr = cadenapararduinoma
nual.c_str();
    std::cout << cstr << std::endl;
#ifdef __arm__
serialPuts(fd, cstr);
#endif
   if (capturarengrane) {
    std::cout << "La captura esta abierta
?: " << cap.isOpened() << std::endl;
      cap >> imagen; //Guardo la imagen d
e la cámara del PC en la matriz
    std::cout << "La imagen esta vacia?"
<< imagen.empty() << std::endl;
    // imshow("Imagen", imagen);
waitKey(1000);
procesarengrane(false);
std::cout << "Si" << std::endl;
  imencode(".png", imagen, buf); //Codifi
co la imagen en PNG y le paso a un Buffer
de memoria
   s = std::string(buf.begin(), buf.end()
); //Transformo el buffer a string
                                    encod
ed = base64_encode(reinterpret_cast<const
unsigned char*> (s.c_str()), s.length());
//Convierto esa String a String en Base64
                                    tipo
= "png";
imencode(".png", roi, buf); //Codifico la
imagen en PNG y le paso a un Buffer de me
moria
  s = std::string(buf.begin(), buf.end())
; //Transformo el buffer a string
  encodedprocesada = base64_encode(reinte
rpret_cast<const unsigned char*> (s.c_str
()), s.length()); //Convierto esa String
a String en Base64
  std::cout << "Si" << std::endl;
  chrono::system_clock::time_point today
= chrono::system_clock::now(); //Obtengo
el tiempo actual
  //Mensajes para enviar
  sio::message::ptr mensajeimagenindividu
al(sio::object_message::create());
  std::map<std::string, sio::message::ptr
>& mapimagenindividual = mensajeimagenind
ividual->get_map();

  tt = chrono::system_clock::to_time_t(to
day); //Convierto a tiempo estandar
  mapimagenindividual.insert(std::make_pa
ir("cadena64", sio::string_message::creat
e(encoded))); //Incluyo el valor de la ca
dena, con un identificador cadena64
  mapimagenindividual.insert(std::make_pa
ir("cadena64procesada", sio::string_messa
ge::create(encodedprocesada))); //Incluyo
el valor de la cadena, con un identificad
or cadena64
  mapimagenindividual.insert(std::make_pa
ir("extension", sio::string_message::crea
te(tipo))); //Incluyo la extension adecua
da
  mapimagenindividual.insert(std::make_pa
ir("fecha", sio::string_message::create(c
time(&tt)))); //Guardo como formato estan
dar compatible con JavaScript
  mapimagenindividual.insert(std::make_pa
ir("errores", sio::string_message::create
(errores))); //Guardo como formato estand
ar compatible con JavaScript
  mapimagenindividual.insert(std::make_pa
ir("numerodientes", sio::int_message::cre
ate(crestastotales.size() + dientesperdid
os))); //Dientes engrane
  mapimagenindividual.insert(std::make_pa
ir("modulo", sio::double_message::create(
modulo))); //Dientes engrane
  mapimagenindividual.insert(std::make_pa
ir("paso", sio::double_message::create(pa
so))); //Dientes engrane
  mapimagenindividual.insert(std::make_pa
ir("diametroexterno", sio::double_message
```

```
::create(2 * radioexterior * constantecon
versionpxacm))); //Dientes engrane
  current_socket->emit("imagen", mensajei
magenindividual); //Envio los datos
  mapimagenindividual.clear();

std::cout << "Fin envio imagenes" << std:
:endl;
                        }
                    }

 std::cout << "Cadena para Equipo 1 mauna
l: ";
cadenapararduinomanual = "";

   datoenvioarduinomanual.printTo(cadenap
ararduinomanual);
   // cadenapararduino += std::string("\0
"); //Fin de cadena
   const char *cstr = cadenapararduinoman
ual.c_str();
  std::cout << cstr << std::endl;
#ifdef __arm__
   serialPuts(fd, cstr);
#endif
   std::cout << "cadenaresetea" << std::e
ndl;
cadenapararduinomanual = "";
std::cout << "cadenaresetea" << cadenapar
arduinomanual << std::endl;
                }
            }

            /* NUEVO EQUIPO*/
 if (mapa.find("equipo2") != mapa.end())
{
  if (mapa.find("equipo2") != mapa.end())
{
  std::cout << "equipo2 : " << mapa.find(
"equipo2")->second->get_int() << std::end
l;
datoenvioarduinoeq2["equipo2"] = mapa.fin
d("equipo2")->second->get_int();
                }
                if (mapa.find("led") != m
apa.end()) {
                    std::cout << "led : "
<< mapa.find("led")->second->get_bool() <
< std::endl;
                            datoenvioardu
inoeq2["led"] = mapa.find("led")->second-
>get_bool();
                }
                if (mapa.find("rgb") != m
apa.end()) {
                    std::cout << "a" << s
td::endl;
                            std::vector<s
io::message::ptr> vectorrgb = mapa.find("
rgb")->second->get_vector();
                            std::cout <<
"Vector rgb: " << vectorrgb.size();
                    if (vectorrgb.size()
== 3) {
                        std::cout << vect
orrgb.at(0)->get_int() << std::endl;
                                    JsonArray
& rgb = datoenvioarduino["rgb"];
                                    int rgb0
= vectorrgb.at(0)->get_int(); // 255
                                    int rgb1
= vectorrgb.at(1)->get_int(); // 255
                                    int rgb2
= vectorrgb.at(2)->get_int(); // 255
                        }
                }
    /*FIN DE NUEVO EQUIPO*/

  std::cout << "Cadena para Equipo 2: ";
  cadenapararduinoeq2 = "";
   datoenvioarduinoeq2.printTo(cadenapara
rduinoeq2);
    cadenapararduinoeq2 += std::string("\
0"); //Fin de cadena
  const char *cstr2 = cadenapararduinoeq2
.c_str();
   std::cout << cstr2 << std::endl;
#ifdef __arm__
   //serialPuts(fd2, cstr2);
#endif
            }

        }

        _lock.unlock(); //Desbloquear la
ejecucion para seguir con el programa
    }));
    current_socket->on("imagen", sio::soc
ket::event_listener_aux([&](string
            const& name, message::ptr con
st& data, bool isAck, message::list & ack
_resp) {

        _lock.lock(); //Bloquear la ejecu
cion hasta realizar una opcion
        std::cout << "Llego una imagen" <
< std::endl;
        _lock.unlock(); //Desbloquear la
ejecucion para seguir con el programa
    }));


    current_socket->on("nuevousuario", si
o::socket::event_listener_aux([&](string
            const& name, message::ptr con
st& data, bool isAck, message::list & ack
_resp) {
        _lock.lock(); //Bloquear la ejecu
cion hasta realizar una opcion
        std::vector<std::shared_ptr < mes
sage>> vectoringresojson = data -> get_ve
ctor(); //Creo vector para el ingreso de
datos
     std::cout << "Inicializando datos" <
< std::endl;

  for (int i = 0; i < vectoringresojson.s
ize(); i++) {
     std::map<std::string, message::ptr>
mapa = vectoringresojson.at(i)->get_map()
;
```

```
    std::cout << "Vector ingreso json a m
apadatos" << std::endl;
            if (mapa.find("motor") != map
a.end()) {
       std::cout << "Motor: " << mapa.fin
d("motor")->second->get_int() << std::end
l;
     datoenvioarduinomotor["motor"] = map
a.find("motor")->second->get_int();

std::cout << "Cadena para Equipo 1 motor:
";
  cadenapararduinomotor = "";
  datoenvioarduinomotor.printTo(cadenapar
arduinomotor);
// cadenapararduino += std::string("\0");
//Fin de cadena
 const char *cstr = cadenapararduinomotor
.c_str();
  std::cout << cstr << std::endl;
#ifdef __arm__
    serialPuts(fd, cstr);
#endif
    std::cout << "cadenaresetea" << std::
endl;
    cadenapararduinomotor = "";
  std::cout << "cadenaresetea" << cadenap
ararduinomotor << std::endl;
     }
  if (mapa.find("dispositivo") != mapa.en
d()) {
    if (mapa.find("dispositivo") != mapa.
end()) {
   std::cout << "dispositivo : " << mapa.
find("dispositivo")->second->get_int() <<
std::endl;
  datoenvioarduino["dispositivo"] = mapa.
find("dispositivo")->second->get_int();
  datoenvioarduinomanual["dispositivo"] =
mapa.find("dispositivo")->second->get_int
();
             }
                if (mapa.find("motor") !=
mapa.end()) {
                    std::cout << "Motor:
" << mapa.find("motor")->second->get_int(
) << std::endl;
                            datoenvioardu
ino["motor"] = mapa.find("motor")->second
->get_int();
                            datoenvioardu
inomanual["motor"] = mapa.find("motor")->
second->get_int();

                }

                if (mapa.find("modofunc")
!= mapa.end()) {
                    std::cout << "modofun
c: " << mapa.find("modofunc")->second->ge
t_bool() << std::endl;
                            modfun = mapa
.find("modofunc")->second->get_bool();

                }
                if (modfun == true) {
                    datoenvioarduino["mod
ofunc"] = mapa.find("modofunc")->second->
get_bool();
                    if (mapa.find("motor"
) != mapa.end()) {
                        std::cout << "Mot
or: " << mapa.find("motor")->second->get_
int() << std::endl;
                            datoenvio
arduino["motor"] = mapa.find("motor")->se
cond->get_int();
                    }

                    if (mapa.find("estado
") != mapa.end()) {
                        std::cout << "est
ado: " << mapa.find("estado")->second->ge
t_int() << std::endl;
                            signed in
t datoestado = mapa.find("estado")->secon
d->get_int();
                            datoenvio
arduino["estado"] = datoestado;
                    }
std::cout << "Cadena para Equipo 1 automa
tico: ";
          cadenapararduino = "";

datoenvioarduino.printTo(cadenapararduino
);
     // cadenapararduino += std::string("
\0"); //Fin de cadena
        const char *cstr = cadenaparardui
no.c_str();
        std::cout << cstr << std::endl;
#ifdef __arm__
     serialPuts(fd, cstr);
#endif
     std::cout << "cadenaresetea" << std:
:endl;
  cadenapararduino = "";
    std::cout << "cadenaresetea" << caden
apararduino << std::endl;


   } else {
    datoenvioarduinomanual["modofunc"] =
mapa.find("modofunc")->second->get_bool()
;
  if (mapa.find("sentidomotor") != mapa.e
nd()) {
  std::cout << "sentidomotor: " << mapa.f
ind("sentidomotor")->second->get_int() <<
std::endl;
   datoenvioarduinomanual["sentidomotor"]
= mapa.find("sentidomotor")->second->get_
int();
    }

  if (mapa.find("direcclasificacion") !=
mapa.end()) {
std::cout << "direcclasificacion: " << ma
pa.find("direcclasificacion")->second->ge
t_int() << std::endl;
```

```
datoenvioarduinomanual["direcclasificacio
n"] = mapa.find("direcclasificacion")->se
cond->get_int();
                    }
  if (mapa.find("umbral") != mapa.end())
{
std::cout << "Umbral IF: " << mapa.find("
umbral")->second->get_int() << std::endl;
datoenvioarduinomanual["umbral"] = mapa.f
ind("umbral")->second->get_int();
                    }
if (mapa.find("capturarengrane") != mapa.
end()) {
std::cout << "capturarengrane: " << mapa.
find("capturarengrane")->second->get_bool
() << std::endl;
datoenvioarduinomanual["capturarengrane"]
= mapa.find("capturarengrane")->second->g
et_bool();
 bool capturarengrane = mapa.find("captur
arengrane")->second->get_bool();
datoenvioarduinomanual.printTo(cadenapara
rduinomanual);
   // cadenapararduino += std::string("\0
"); //Fin de cadena
  const char *cstr = cadenapararduinomanu
al.c_str();
  std::cout << cstr << std::endl;
#ifdef __arm__
  serialPuts(fd, cstr);
#endif
     if (capturarengrane) {
      std::cout << "La captura esta abier
ta?: " << cap.isOpened() << std::endl;
  cap >> imagen; //Guardo la imagen de la
cámara del PC en la matriz
  std::cout << "La imagen esta vacia?" <<
imagen.empty() << std::endl;
  //  imshow("Imagen", imagen);
  waitKey(1000);
  procesarengrane(false);

  std::cout << "Si" << std::endl;
  imencode(".png", imagen, buf); //Codifi
co la imagen en PNG y le paso a un Buffer
de memoria
  s = std::string(buf.begin(), buf.end())
; //Transformo el buffer a string
  encoded = base64_encode(reinterpret_cas
t<const unsigned char*> (s.c_str()), s.le
ngth()); //Convierto esa String a String
en Base64
  tipo = "png";
  imencode(".png", roi, buf); //Codifico
la imagen en PNG y le paso a un Buffer de
memoria
  s = std::string(buf.begin(), buf.end())
; //Transformo el buffer a string
  encodedprocesada = base64_encode(reinte
rpret_cast<const unsigned char*> (s.c_str
()), s.length()); //Convierto esa String
a String en Base64
  std::cout << "Si" << std::endl;
  chrono::system_clock::time_point today
= chrono::system_clock::now(); //Obtengo
el tiempo actual
  //Mensajes para enviar
  sio::message::ptr mensajeimagenindividu
al(sio::object_message::create());
  std::map<std::string, sio::message::ptr
>& mapimagenindividual = mensajeimagenind
ividual->get_map();

  tt = chrono::system_clock::to_time_t(to
day); //Convierto a tiempo estandar
  mapimagenindividual.insert(std::make_pa
ir("cadena64", sio::string_message::creat
e(encoded))); //Incluyo el valor de la ca
dena, con un identificador cadena64
  mapimagenindividual.insert(std::make_pa
ir("cadena64procesada", sio::string_messa
ge::create(encodedprocesada))); //Incluyo
el valor de la cadena, con un identificad
or cadena64
  mapimagenindividual.insert(std::make_pa
ir("extension", sio::string_message::crea
te(tipo))); //Incluyo la extension adecua
da
  mapimagenindividual.insert(std::make_pa
ir("fecha", sio::string_message::create(c
time(&tt)))); //Guardo como formato estan
dar compatible con JavaScript
  mapimagenindividual.insert(std::make_pa
ir("errores", sio::string_message::create
(errores))); //Guardo como formato estand
ar compatible con JavaScript
  mapimagenindividual.insert(std::make_pa
ir("numerodientes", sio::int_message::cre
ate(crestastotales.size() + dientesperdid
os))); //Dientes engrane
  mapimagenindividual.insert(std::make_pa
ir("modulo", sio::double_message::create(
modulo))); //Dientes engrane
  mapimagenindividual.insert(std::make_pa
ir("paso", sio::double_message::create(pa
so))); //Dientes engrane
  mapimagenindividual.insert(std::make_pa
ir("diametroexterno", sio::double_message
::create(2 * radioexterior * constantecon
versionpxacm))); //Dientes engrane
  current_socket->emit("imagen", mensajei
magenindividual); //Envio los datos
  mapimagenindividual.clear();
                        }
                    }

std::cout << "Cadena para Equipo 1 maunal
: ";
   cadenapararduinomanual = "";

   datoenvioarduinomanual.printTo(cadenap
ararduinomanual);
// cadenapararduino += std::string("\0");
//Fin de cadena
const char *cstr = cadenapararduinomanual
.c_str();
 std::cout << cstr << std::endl;
#ifdef __arm__
serialPuts(fd, cstr);
#endif
std::cout << "cadenaresetea" << std::endl
;
```

```
cadenapararduinomanual = "";
 std::cout << "cadenaresetea" << cadenapa
rarduinomanual << std::endl;
                }

       /* NUEVO EQUIPO*/
   if (mapa.find("equipo2") != mapa.end()
) {
     if (mapa.find("equipo2") != mapa.end
()) {
   std::cout << "equipo2 : " << mapa.find
("equipo2")->second->get_int() << std::en
dl;
      datoenvioarduinoeq2["equipo2"] = ma
pa.find("equipo2")->second->get_int();
                    }
if (mapa.find("led") != mapa.end()) {
  std::cout << "led : " << mapa.find("led
")->second->get_bool() << std::endl;
    datoenvioarduinoeq2["led"] = mapa.fin
d("led")->second->get_bool();
                    }
   if (mapa.find("rgb") != mapa.end()) {
     std::cout << "a" << std::endl;
        std::vector<sio::message::ptr> ve
ctorrgb = mapa.find("rgb")->second->get_v
ector();
  std::cout << "Vector rgb: " << vectorrg
b.size();
  if (vectorrgb.size() == 3) {
   std::cout << vectorrgb.at(0)->get_int(
) << std::endl;
  JsonArray& rgb = datoenvioarduino["rgb"
];
  int rgb0 = vectorrgb.at(0)->get_int();
// 255
  int rgb1 = vectorrgb.at(1)->get_int();
// 255
  int rgb2 = vectorrgb.at(2)->get_int();
// 255
                        }
                    }
/*FIN DE NUEVO EQUIPO*/
std::cout << "Cadena para Equipo 2: ";
    cadenapararduinoeq2 = "";
datoenvioarduinoeq2.printTo(cadenaparardu
inoeq2);
   cadenapararduinoeq2 += std::string("\0
"); //Fin de cadena
   const char *cstr2 = cadenapararduinoeq
2.c_str();
std::cout << cstr2 << std::endl;
#ifdef __arm__
  //serialPuts(fd2, cstr2);
#endif
                }

            }
        }

        _lock.unlock(); //Desbloquear la
ejecucion para seguir con el programa
    }));
}

//Separar string con delimitador
// manejamos por interfaz de usuario ejem
plo: motor,225, led,0.......

std::vector<std::string> split(std::strin
g strToSplit, char delimeter) {
    std::stringstream ss(strToSplit);
    std::string item;
    std::vector<std::string> splittedStri
ngs;
    while (std::getline(ss, item, delimet
er)) {
   splittedStrings.push_back(item);
    }
    return splittedStrings;
}

void ayuda() {

    HIGHLIGHT("Ayuda, no hay suficientes
argumentos");
    std::cout << "Opciones validas" << st
d::endl;
    std::cout << "--s <ip> especifica la
dirección IP servidor" << std::endl;
    std::cout << "--p <puerto> especifica
puerto del servidor" << std::endl;
    std::cout << "--ps <puertoserial> esp
ecifica la direccion del puerto donde se
encuentra conectado la comunicacion seria
l" << std::endl;
    std::cout << "--c <id camara> especif
ica el numero entero que identifica la cá
mara en el openCV " << std::endl;

}

int main(int argc, const char* args[]) {
    if (argc > 2) { //Para cambiar el pue
rto y el servidor desde el terminal
        for (int j = 0; j < argc; j++) {

            if (strcmp(args[j], "--s") ==
0) {
                std::cout << "Cambio serv
idor" << std::endl;
                if ((j + 1) <= argc) serv
idorIP = std::string(args[j + 1]);
                else ayuda();
                std::cout << "Servidor IP
: " << servidorIP << std::endl;
            } else if (strcmp(args[j], "-
-p") == 0) {
                std::cout << "Cambio puer
to" << std::endl;
                if ((j + 1) <= argc) puer
to = std::string(args[j + 1]);
                else ayuda();
            } else if (strcmp(args[j], "-
-ps") == 0) {
                std::cout << "Cambio puer
to serial" << std::endl;
                if ((j + 1) <= argc) puer
toserial = std::string(args[j + 1]);
                else ayuda();
            } else if (strcmp(args[j], "-
-c") == 0) {
```

```
                std::cout << "Cambio  de
camara" << std::endl;
                if ((j + 1) <= argc) idca
mara = atoi(args[j + 1]);
                else ayuda();
            }

        }
    } else ayuda();
#ifdef __arm__
    if ((fd = serialOpen(puertoserial.c_s
tr(), 115200)) < 0) {
        fprintf(stderr, "No se puede abri
r el dispositivo serial EQUIPO1, revise l
os permisos!: %s\n", strerror(errno));
        return -1; //ttyUSB0" Regresa con
error
    }
    /* Diria que no es recomendable manda
rle a la misma direccion
    if ((fd2 = serialOpen("/dev/ttyUSB0",
115200)) < 0) {
        fprintf(stderr, "No se puede abri
r el dispositivo serial EQUIPO1, revise l
os permisos!: %s\n", strerror(errno));
        return 1; //ttyUSB0"
    }/*

    /*  if ((fd2 = serialOpen("/dev/ttyAC
M0", 115200)) < 0) {

          fprintf(stderr, "No se puede ab
rir el EQUIPO 2 serial, revise los permis
os!: %s\n", strerror(errno));
          return 1;
      }*/

    if (wiringPiSetup() == -1) {
        // fprintf(stdout, "No se puede i
nicializar wiringPi: %s\n", strerror(errn
o));
        return -11;
    }
#endif

    sio::client h; //Creación de un nuevo
cliente de Sockets.io
    connection_listener l(h); //Añado al
cliente un listener para obtener eventos
    h.set_open_listener(std::bind(&connec
tion_listener::on_connected, &l));
    h.set_close_listener(std::bind(&conne
ction_listener::on_close, &l, std::placeh
olders::_1));
    h.set_fail_listener(std::bind(&connec
tion_listener::on_fail, &l));
    std::cout << "Corriendo sobre el serv
idor " + servidorIP + ":" + puerto << std
::endl;
    h.connect("http://" + servidorIP + ":
" + puerto); //Conexión  al servidor Web

    _lock.lock();
    if (!connect_finish) {
        _cond.wait(_lock);
    }
    _lock.unlock();
    current_socket = h.socket(); //Socket
para inicio de comunicacion

    std::vector<std::string> cadenasepara
da, extensionarchivo; //Variables para se
parar string, una para separar la entrada
de texto, otra para obtener la extension
del archivo
    sio::message::ptr mensajeunico(sio::o
bject_message::create()); //Objetos para
enviar datos en Formato JSON
    sio::message::ptr mensajeimagen(sio::
object_message::create());

    std::map<std::string, sio::message::p
tr>& map = mensajeunico->get_map(); //Pun
teros hacia los objetos anteriores
    std::map<std::string, sio::message::p
tr>& mapimagen = mensajeimagen->get_map()
;

    array_message::ptr mensajetotal(sio::
array_message::create()); //Crear un arra
y de objetos, para enviar varios datos a
la vez
    std::vector<std::shared_ptr < sio::me
ssage>> &vectortotal = mensajetotal->get_
vector(); //Variable estandar con puntero
al anterior objeto array_message

    eventos(); //mandar a llamar una vez,
obligatorio
    const String rutared = "parametrosred
.xml";
    /*red = cv::ml::ANN_MLP::load(rutared
);
    if (red->empty()) {
        std::cout << "No hay red" << std:
:endl;
        return -1;
    }*/
    datosprueba = Mat(1, entradared.heigh
t * entradared.width, CV_32F, Scalar(0));
    cap.open(idcamara);
    if (!cap.isOpened()) {
        std::cout << "\033[1;31m No esta
conectada la camara al dispositivo, conec
tela por favor \033[0m" << std::endl;
        return -1;
    }
#ifdef __arm__
    std::cout << "Corriendo sobre un disp
ostivo arm" << std::endl;
    std::string cadenatotaldesdearduino =
"";
    std::string cadenatotaldesdearduinoeq
2 = "";

    bool sensorpresencia = false;
    char c;
    char z;
    bool llegodatos = false;
    bool llegodatoseq2 = false;
    while (1) {
        cap >> imagen;
```

```
        llegodatos = false;
        while (serialDataAvail(fd) > 0) {
            c = serialGetchar(fd);
            std::cout << c;
            if (c == '{') {
                iniciocadena = true;
                cadenatotaldesdearduino =
"";
            }
            if (iniciocadena) {
                cadenatotaldesdearduino +
= std::string(1, c);
                if (c == '}') {
                    fincadena = true;
                    std::cout << "Caracte
r }" << std::endl;
                    break;
                }
            }
        }
        if (iniciocadena && fincadena) {
            iniciocadena = false, fincade
na = false;
            std::cout << "Procesando JSON
" << std::endl;
            llegodatos = false;
            JsonObject& datojson = Buffer
entrada.parseObject(cadenatotaldesdeardui
no);
            std::cout << "Cadena que lleg
a Arduino: " << cadenatotaldesdearduino <
< std::endl;
            cadenatotaldesdearduino = "";
            if (datojson.success()) {
                map.clear();
                for (auto kv : datojson)
{
                    if (strcmp(kv.key, ("
presencia")) == 0) {
                        map.insert(std::m
ake_pair("presencia",
                                   sio::bool
_message::create(kv.value.as<bool>())));
                        if (kv.value.as<b
ool>()) {
                            std::cout <<
"Presencia: true" << std::endl;
                            sensorpresenc
ia = true;
                        } else {
                            std::cout <<
"Presencia: false" << std::endl;
                            sensorpresenc
ia = false;
                        }
                    }
                    if (strcmp(kv.key, ("
estado")) == 0) {
                        map.insert(std::m
ake_pair("estado",
                                   sio::int_
message::create(kv.value.as<signed int>()
)));
                    }
                }
                vectortotal.push_back(men
sajeunico);
                current_socket->emit("dat
os", mensajetotal);
                vectortotal.clear();
            }
        }
        if (sensorpresencia) {
            //Cuando la camara tiene comp
ensacion automatica de exposicion captura
mos siete imagenes hasta que se estabilic
e
            for (int k = 0; k < 4; k++) {
                cap >> imagen;
            }
            int resultado = procesarengra
ne(true);
            imencode(".png", imagen, buf)
; //Codifico la imagen en PNG y le paso a
un Buffer de memoria
            s = std::string(buf.begin(),
buf.end()); //Transformo el buffer a stri
ng
            encoded = base64_encode(reint
erpret_cast<const unsigned char*> (s.c_st
r()), s.length()); //Convierto esa String
a String en Base64
            tipo = "png";
            imencode(".png", roi, buf); /
/Codifico la imagen en PNG y le paso a un
Buffer de memoria
            s = std::string(buf.begin(),
buf.end()); //Transformo el buffer a stri
ng
            encodedprocesada = base64_enc
ode(reinterpret_cast<const unsigned char*
> (s.c_str()), s.length()); //Convierto e
sa String a String en Base64
            mapimagen.insert(std::make_pa
ir("cadena64", sio::string_message::creat
e(encoded))); //Incluyo el valor de la ca
dena, con un identificador cadena64
            chrono::system_clock::time_po
int today = chrono::system_clock::now();
//Obtengo el tiempo actual
            tt = chrono::system_clock::to
_time_t(today); //Convierto a tiempo esta
ndar
            std::cout << "today is: " <<
ctime(&tt);
            mapimagen.insert(std::make_pa
ir("cadena64", sio::string_message::creat
e(encoded))); //Incluyo el valor de la ca
dena, con un identificador cadena64
            mapimagen.insert(std::make_pa
ir("cadena64procesada", sio::string_messa
ge::create(encodedprocesada))); //Incluyo
el valor de la cadena, con un identificad
or cadena64
            mapimagen.insert(std::make_pa
ir("extension", sio::string_message::crea
te(tipo))); //Incluyo la extension adecua
da
            mapimagen.insert(std::make_pa
ir("fecha", sio::string_message::create(c
```

```
time(&tt))); //Guardo como formato estan
dar compatible con JavaScript
            mapimagen.insert(std::make_pa
ir("errores", sio::string_message::create
(errores))); //Guardo como formato estand
ar compatible con JavaScript
            mapimagen.insert(std::make_pa
ir("numerodientes", sio::int_message::cre
ate(crestastotales.size() + dientesperdid
os))); //Dientes engrane
            mapimagen.insert(std::make_pa
ir("modulo", sio::double_message::create(
modulo))); //Dientes engrane
            mapimagen.insert(std::make_pa
ir("paso", sio::double_message::create(pa
so))); //Dientes engrane
            mapimagen.insert(std::make_pa
ir("diametroexterno", sio::double_message
::create(2 * radioexterior * constantecon
versionpxacm))); //Dientes engrane

            current_socket->emit("imagen"
, mensajeimagen); //Envio los datos
            mapimagen.clear();
            JsonObject& datoenvioarduinov
id = Buffersalida.createObject();

            datoenvioarduinovid["imagenpr
ocesada"] = true;
            std::cout << "Resultado: " <<
resultado << endl;
            if (resultado == true) datoen
vioarduinovid["clasificador"] = true;
            else datoenvioarduinovid["cla
sificador"] = false;
            datoenvioarduinovid.printTo(c
adenapararduinovid);
            cadenapararduinovid += std::s
tring("\0"); //Caracter terminacion
            const char *cstr = cadenapara
rduinovid.c_str();
            std::cout << "Cadena de image
n procesada " << cstr << std::endl;
            serialPuts(fd, cstr); //Envio
que realizo la vision
            std::cout << "Fin Vision" <<
std::endl;
            sensorpresencia = false;
            cstr = "";
            cadenapararduinovid = "";
        }
        /* if (capturarengrane == true) {

            //Cuando la camara tiene comp
ensacion automatica de exposicion captura
mos siete imagenes hasta que se estabilic
e
            for (int k = 0; k < 4; k++) {
                cap >> imagen;
            }
            int resultado = procesarengra
ne(true);
            imencode(".png", imagen, buf)
; //Codifico la imagen en PNG y le paso a
un Buffer de memoria
            s = std::string(buf.begin(),
buf.end()); //Transformo el buffer a stri
ng
            encoded = base64_encode(reint
erpret_cast<const unsigned char*> (s.c_st
r()), s.length()); //Convierto esa String
a String en Base64
            tipo = "png";
            imencode(".png", roi, buf); /
/Codifico la imagen en PNG y le paso a un
Buffer de memoria
            s = std::string(buf.begin(),
buf.end()); //Transformo el buffer a stri
ng
            encodedprocesada = base64_enc
ode(reinterpret_cast<const unsigned char*
> (s.c_str()), s.length()); //Convierto e
sa String a String en Base64
            mapimagen.insert(std::make_pa
ir("cadena64", sio::string_message::creat
e(encoded))); //Incluyo el valor de la ca
dena, con un identificador cadena64
            chrono::system_clock::time_po
int today = chrono::system_clock::now();
//Obtengo el tiempo actual
            tt = chrono::system_clock::to
_time_t(today); //Convierto a tiempo esta
ndar
            std::cout << "today is: " <<
ctime(&tt);
            mapimagen.insert(std::make_pa
ir("cadena64", sio::string_message::creat
e(encoded))); //Incluyo el valor de la ca
dena, con un identificador cadena64
            mapimagen.insert(std::make_pa
ir("cadena64procesada", sio::string_messa
ge::create(encodedprocesada))); //Incluyo
el valor de la cadena, con un identificad
or cadena64
            mapimagen.insert(std::make_pa
ir("extension", sio::string_message::crea
te(tipo))); //Incluyo la extension adecua
da
            mapimagen.insert(std::make_pa
ir("fecha", sio::string_message::create(c
time(&tt)))); //Guardo como formato estan
dar compatible con JavaScript
            mapimagen.insert(std::make_pa
ir("errores", sio::string_message::create
(errores))); //Guardo como formato estand
ar compatible con JavaScript
            mapimagen.insert(std::make_pa
ir("numerodientes", sio::int_message::cre
ate(crestastotales.size() + dientesperdid
os))); //Dientes engrane
            mapimagen.insert(std::make_pa
ir("modulo", sio::double_message::create(
modulo))); //Dientes engrane
            mapimagen.insert(std::make_pa
ir("paso", sio::double_message::create(pa
so))); //Dientes engrane
            mapimagen.insert(std::make_pa
ir("diametroexterno", sio::double_message
::create(2 * radioexterior * constantecon
versionpxacm))); //Dientes engrane
```

```
            current_socket->emit("imagen
", mensajeimagen); //Envio los datos
             mapimagen.clear();
             capturaengrane = false;
         }*/
        //NUEVO DISPOSITIVO
        llegodatoseq2 = false;
        /*while (serialDataAvail(fd2) > 0
) {
            z = serialGetchar(fd2);
            std::cout << z;
            if (z == '{') {
                iniciocadenaeq2 = true;
                cadenatotaldesdearduinoeq
2 = "";
            }
            if (iniciocadena) {
                cadenatotaldesdearduinoeq
2 += std::string(1, z);
                if (c == '}') {
                    fincadenaeq2 = true;
                    std::cout << "Caracte
r }" << std::endl;
                    break;
                }
            }
        }*/
        if (iniciocadenaeq2 && fincadenae
q2) {
            iniciocadenaeq2 = false, finc
adenaeq2 = false;
            std::cout << "Procesando JSON
" << std::endl;
            llegodatoseq2 = false;
            JsonObject& datojson = Buffer
entradaeq2.parseObject(cadenatotaldesdear
duinoeq2);
            std::cout << "Cadena que lleg
a Arduino: " << cadenatotaldesdearduinoeq
2 << std::endl;
        cadenatotaldesdearduinoeq2 = "";
            if (datojson.success()) {
                map.clear();
     for (auto kv : datojson) {
  if (strcmp(kv.key, ("led")) == 0) {
   map.insert(std::make_pair("led",
   sio::bool_message::create(kv.value.as<
bool>())));
     if (kv.value.as<bool>()) {
     std::cout << "led: true" << std::end
l;
    } else {
  std::cout << "led: false" << std::endl;

                }
             }
       }
     vectortotal.push_back(mensajeunico);
   current_socket->emit("datos", mensajet
otal);
       vectortotal.clear();
            }
        }
    }
#else
    std::cout << "Corriendo en un computa
dor x86-64" << std::endl;
    HIGHLIGHT("Manual: Ponga el identific
ador seguido del valor, ejemplo:");
    EM(" motor,255; estado, 0; presencia,
0; clasificador,0; Valores permitidos:mot
or(0-255), clasificador(0,1), presencia(1
,0), led(1,0), imagen ,ruta/a/imagen.gif;
");
    for (std::string line; std::getline(s
td::cin, line);) {
        cap >> imagen;
        if (line.length() > 0) {
            if (line == "$exit") {
                std::cout << "Salir" << s
td::endl;
                //  break;
            } else if (line.length() > 4
&& line.length() < 300) { //Si tiene un t
amaño estándar
                cadenaseparada = split(li
ne, ','); //Separe las opciones
                if (cadenaseparada.size()
% 2 == 0) { //Si las opciones son divisib
les para dos quiere decir que se puso el
identificador y el valor
                    for (int i = 0; i < c
adenaseparada.size(); i += 2) {
if (cadenaseparada.at(i) == "dispositivo"
) { //Si es el identificador motor
          datos = true;
   map.insert(std::make_pair("dispositivo
",
  sio::int_message::create(stoi(cadenasep
arada.at(i + 1))))); //Crea el Par nombre
:actuadorMotor
 std::cout << "Dispositivo" << std::endl;
 } else if (cadenaseparada.at(i) == "moto
r") {
      datos = true;
    map.insert(std::make_pair("motor",
  sio::int_message::create(stoi(cadenasep
arada.at(i + 1)))));
 std::cout << "Motor" << std::endl;
                        } else if (cadena
separada.at(i) == "presencia") {
     datos = true;
if (cadenaseparada.at(i + 1) == "true") {
map.insert(std::make_pair("presencia",
      sio::bool_message::create(true)));
                            } else {
  map.insert(std::make_pair("presencia",
      sio::bool_message::create(false)));
                            }
  std::cout << "Presencia" << std::endl;
                        } else if (cadena
separada.at(i) == "estado") {
   datos = true;
 map.insert(std::make_pair("estado",
  sio::int_message::create(stoi(cadenasep
arada.at(i + 1)))));
 std::cout << "Estado" << std::endl;
      } else if (cadenaseparada.at(i) ==
"imagenprocesada") {
    datos = true;
```

```
  if (cadenaseparada.at(i + 1) == "true")
{
  map.insert(std::make_pair("imagenproces
ada",
      sio::bool_message::create(true)));
  } else {
   map.insert(std::make_pair("imagenproce
sada",
      sio::bool_message::create(false)));
       }
      } else if (cadenaseparada.at(i) ==
"clasificador") {
     datos = true;
if (cadenaseparada.at(i + 1) == "true") {
map.insert(std::make_pair("clasificador",
      sio::bool_message::create(true)));
     } else {
  map.insert(std::make_pair("clasificador
",
      sio::bool_message::create(false)));
          }
     } else if (cadenaseparada.at(i) == "
imagen") { //Si es imagen
    if (cadenaseparada.at(i + 1) == "0")
{ //Si es cero saca imagen de la camara
     std::cout << "camara" << std::endl;
       //Saque imagen de la camara del Co
mputador
  cap.open(0);
     cap >> imagen; //Guardo la imagen de
la cámara del PC en la matriz
    procesarengrane(false);
  imencode(".png", imagen, buf); //Codifi
co la imagen en PNG y le paso a un Buffer
de memoria
  s = std::string(buf.begin(), buf.end())
; //Transformo el buffer a string
  encoded = base64_encode(reinterpret_cas
t<const unsigned char*> (s.c_str()), s.le
ngth()); //Convierto esa String a String
en Base64
       tipo = "png";
    imencode(".png", roi, buf); //Codific
o la imagen en PNG y le paso a un Buffer
de memoria
                                s = std::
string(buf.begin(), buf.end()); //Transfo
rmo el buffer a string
                                encodedpr
ocesada = base64_encode(reinterpret_cast<
const unsigned char*> (s.c_str()), s.leng
th()); //Convierto esa String a String en
Base64
  cap.release();
        } else {
   std::cout << "Archivo " << std::endl;
        std::cout << cadenaseparada.at(i
+ 1) << std::endl;
  tipo = split(cadenaseparada.at(i + 1),
'.').at(1); //Obtengo la extension del ar
chivo
   std::cout << "Tipo: " << tipo << endl;
   if (tipo == "png" || tipo == "jpg" ||
tipo == "jpeg" || tipo == "bmp") {
  imagen = imread(cadenaseparada.at(i + 1
));
  if (imagen.empty()) {
      std::cout << "Mala ruta de la image
n" << std::endl;
      return -1;
  }
  procesarengrane(false);
  imencode(".png", imagen, buf); //Codifi
co la imagen en PNG y le paso a un Buffer
de memoria
  s = std::string(buf.begin(), buf.end())
; //Transformo el buffer a string
  encoded = base64_encode(reinterpret_cas
t<const unsigned char*> (s.c_str()), s.le
ngth()); //Convierto esa String a String
en Base64
  tipo = "png";
  imencode(".png", roi, buf); //Codifico
la imagen en PNG y le paso a un Buffer de
memoria
  s = std::string(buf.begin(), buf.end())
; //Transformo el buffer a string
  encodedprocesada = base64_encode(reinte
rpret_cast<const unsigned char*> (s.c_str
()), s.length()); //Convierto esa String
a String en Base64
                                } else {
  ifstream in(cadenaseparada.at(i + 1), i
os::binary); //Abro el archivo en forma b
inaria, en la ruta especificada
  in.seekg(0, ios::end); //Itero sobre to
do el archivo
  int iSize = in.tellg(); //Obtengo el ta
maño del Archivo
  in.seekg(0, ios::beg);
  std::vector<char> pBuff(iSize); //Creo
un buffer del tamaño del archivo
                                if (i
Size > 0)
                                  i
n.read(&pBuff[0], iSize); //Pongo los dat
os de la imagen en el buffer
                                in.cl
ose();
                                std::
string s(pBuff.begin(), pBuff.end()); //C
reo un String con todos los elementos del
Buffer
                                encod
ed = base64_encode(reinterpret_cast<const
unsigned char*> (s.c_str()), s.length());
///Convierto la cadena a un String de Bas
e64
                                }

                            }
  chrono::system_clock::time_point today
= chrono::system_clock::now(); //Obtengo
el tiempo actual
   tt = chrono::system_clock::to_time_t(t
oday); //Convierto a tiempo estandar
  mapimagen.insert(std::make_pair("cadena
64", sio::string_message::create(encoded)
)); //Incluyo el valor de la cadena, con
un identificador cadena64
   mapimagen.insert(std::make_pair("caden
a64procesada", sio::string_message::creat
```

```
e(encodedprocesada))); //Incluyo el valor de la cadena, con un identificador cadena 64
  mapimagen.insert(std::make_pair("extension", sio::string_message::create(tipo))); //Incluyo la extension adecuada
   mapimagen.insert(std::make_pair("fecha", sio::string_message::create(ctime(&tt)))); //Guardo como formato estandar compatible con JavaScript
  mapimagen.insert(std::make_pair("errores", sio::string_message::create(errores))); //Guardo como formato estandar compatible con JavaScript
   mapimagen.insert(std::make_pair("numerodientes", sio::int_message::create(crestastotales.size() + dientesperdidos))); //Dientes engrane
  mapimagen.insert(std::make_pair("modulo", sio::double_message::create(modulo))); //Dientes engrane
  mapimagen.insert(std::make_pair("paso", sio::double_message::create(paso))); //Dientes engrane
                            mapimagen.insert(std::make_pair("diametroexterno", sio::double_message::create(2 * radioexterior * constanteconversionpxacm))); //Dientes engrane
                            current_socket->emit("imagen", mensajeimagen); //Envio los datos
                            mapimagen.clear();
                        }
                    }
                }
                if (datos) {
  vectortotal.push_back(mensajeunico); //Ingresa al vector como ultimo de la cola
                    std::cout << "*======sacando valores del vector final=======*" << std::endl; //Si hay valores en el vector, llene
                    std::cout << "tamaño: " << vectortotal.size() << std::endl;
                    current_socket->emit("datos", mensajetotal); //Envie los datos en un solo vector
                    vectortotal.clear();
                    map.clear(); //Reinicio las variables
                    datos = false;
                }
                cadenaseparada.clear();
                _lock.lock();
                _lock.unlock();
            } else {
                vectortotal.clear();
                map.clear();
                cadenaseparada.clear();
            }
        }
    }
#endif
    h.close(); //Cerrar servidor
    return 0;
```

ANEXO 6 – ALGORITMO DE VISIÓN Y PROCESADO DE ENGRANANJES

```
/*
 * File:   Vision.h
 * Author: Fernando Jácome
 * Created on October 3, 2018, 5:26 AM
 */
#ifndef VISION_H
#define VISION_H

#include <iostream>
#include <opencv2/core.hpp>             /
/Librerias de OpenCV para el procesamient
o
#include <opencv2/imgproc.hpp>
#include <opencv2/highgui/highgui.hpp>
#include <opencv2/ml/ml.hpp>
#include <math.h>
#include <vector>
#include <opencv2/highgui.hpp>
#include <opencv2/core/hal/interface.h>
#include <opencv2/core/types.hpp>
//Variables de OpenCV
cv::Mat imagencontours;
cv::Mat imagengris;
cv::Mat canny_output;
cv::Mat roi, roired;
cv::Mat datosprueba;
cv::Mat salida(1, 1, CV_32FC1); //Datos p
ara el procesamiento red
cv::Mat lineasexternas;
cv::Point2f puntoanterior;
cv::Mat lineasinternas;

std::vector<std::vector<cv::Point> > cres
tastotales, vallestotales;
std::vector<cv::Point> cresta, valle;
std::vector<std::vector<cv::Point> > cont
ours;
std::vector<cv::Point> circulointerno;
std::vector<cv::Vec4i> hierarchy;
std::vector<std::vector<cv::Point> > cont
ours_poly;
int kernelsize = 3;
int tope = 180;
cv::Rect rectaobjetivo;
int offset = 7;
int aciertos = 0;
float distanciaminima = 1000;
float distanciacentropunto = 0;
cv::Point2f centrocirculo;
cv::Point menorradio;
float radioexterior;
int puntosadd = 0;
cv::Point2f centrocirculointerno;
float radiocirculointerno;
float modulo = 0.0;
float paso = 0.0;
cv::Mat imagen;
std::string errores = "";
int numeroerrores = 0;
```

```
#ifdef __arm__
cv::VideoCapture cap(0);
#else
#define mostrarcapturas
cv::VideoCapture cap(1);
#endif
int numerocrestas = 0;
int numerovalles = 0;
int indicemayor = 0; //Indice del contorn
o mayor
int areamayor = 0;
int area = 0;
double areapromediovalles = 0.0, areaprom
ediocrestas = 0.0;
double areacresta = 0.0, areavalle = 0;
double desviacionestandarcresta = 0.0, de
sviacionestandarvalle = 0.0;
//cv::Ptr<cv::ml::ANN_MLP> red; //Puntero
de la red
cv::Size entradared(32, 32); //Entrada de
32x32 a la red
int dientestotales = 0;
int dientesperdidos = 0;
float constanteconversionpxacm = 0.208333
3;
float limitesuperior = 0, limiteinferior
= 0;
float sumatotalcircunferencia = 0;

struct areasrectas {
    double area = 0.0;
    cv::Rect Recta;
    double anguloentrecrestas = 0.0;
    cv::Point2f centroide;
};

void preprocesamiento() {
    //Inicializacion de variables
    crestastotales.clear();
    vallestotales.clear();
    contours.clear();
    hierarchy.clear();
    areamayor = 0;
    indicemayor = 0;
    area = 0;
    numeroerrores = 0;
    dientesperdidos = 0;
    sumatotalcircunferencia = 0;
    imagencontours = cv::Mat(imagen.rows,
imagen.cols, CV_8UC3, cv::Scalar(0, 0, 0)
);
    lineasexternas = cv::Mat(imagen.rows,
imagen.cols, CV_8UC3, cv::Scalar(0, 0, 0)
);
    lineasinternas = cv::Mat(imagen.rows,
imagen.cols, CV_8UC3, cv::Scalar(0, 0, 0)
);
    roi = cv::Mat(imagen.rows, imagen.col
s, CV_8UC3, cv::Scalar(0));
```

```
    errores = "";
    numeroerrores = 0;
    imagengris.release();
    cv::cvtColor(imagen, imagengris, cv::
COLOR_BGR2GRAY); //Transformo a escala de
grises
    normalize(imagengris, imagengris, 0,
255, cv::NORM_MINMAX, -1); //Normalizo la
s imagenes para compensar luces y sombras
    medianBlur(imagengris, imagengris, ke
rnelsize); //difuminacion de la imagen
    cv::threshold(imagengris, imagengris,
tope, 255, cv::THRESH_OTSU | cv::THRESH_B
INARY); //Umbralizacion OTSU de la imagen
    cv::Mat kernelerosion = cv::getStruct
uringElement(cv::MORPH_CROSS, cv::Size(3,
3));
    cv::erode(imagengris, imagengris, ker
nelerosion); //Erosion para eliminar punt
os cercanos
    cv::findContours(imagengris, contours
, hierarchy, CV_RETR_EXTERNAL, cv::CHAIN_
APPROX_NONE, cv::Point(0, 0)); //Buscar c
ontornos
    contours_poly.resize(contours.size())
;

    //Hallar el contorno con el indice ma
yor para procesearlo
    for (int i = 0; i < contours.size();
i++) {
        area = contourArea(contours[i]);
        if (area > areamayor) {
            areamayor = area;
            indicemayor = i;
        }
    }
    cv::approxPolyDP(contours[indicemayor
], contours_poly[indicemayor], 3, true);
//Aproximo un poligono al contorno
    cv::minEnclosingCircle(contours[indic
emayor], centrocirculo, radioexterior); /
/Busco el circulo que se aproxima al cont
orno externo
    rectaobjetivo = cv::boundingRect(cv::
Mat(contours_poly[indicemayor])); //Busco
la recta que contiene al contorno
    cv::drawContours(imagencontours, cont
ours, indicemayor, cv::Scalar(255, 255, 2
55), -1);
    roi = imagencontours(rectaobjetivo);
}

/*
float clasificar() {
    resize(roi, roired, entradared, 0, 0,
cv::INTER_NEAREST);
    int contador = 0;
    for (int i = 0; i < roired.rows; i++)
{
        for (int j = 0; j < roired.cols;
j++) {
            datosprueba.at<float>(0, cont
ador) = roired.at<uchar>(i, j) / 255.0;
            contador += 1;
        }
    }
    red->predict(datosprueba, salida);
    return salida.at<float>(0, 0);
}*/

bool procesarengrane() {
    preprocesamiento();
    if (rectaobjetivo.x == 0 || rectaobje
tivo.x + rectaobjetivo.width == imagen.co
ls) {
        std::cout << "Imagen no centrada,
centrando..." << std::endl;
    }
#ifdef __arm__
    while (rectaobjetivo.x == 0 || rectao
bjetivo.x + rectaobjetivo.width == imagen
.cols) { //Si el contorno mayor esta topa
ndo en cero o esta topando en el borde ma
yor en y de la imagen
        std::cout << "Posicion y: " << re
ctaobjetivo.y << " Posicion x: " << recta
objetivo.x << std::endl;
        std::cout << " Posicion final: "
<< rectaobjetivo.x + rectaobjetivo.width
<< " ancho " << imagen.cols << std::endl;
        if (rectaobjetivo.x == 0) {
            serialPutchar(fd, 'a');
            std::cout << "Motor para adel
ante" << std::endl;
        } else if (rectaobjetivo.x + rect
aobjetivo.width == imagen.cols) {
            serialPutchar(fd, 'd');
            std::cout << "Motor para atra
s" << std::endl;
        }
        while (!serialDataAvail(fd)) {
        } // Espero hasta que haya datos
en el puerto serial
        char dato = serialGetchar(fd); //
Variable para que se transfiera del buffe
r el dato
        std::cout << "Dato bucle while: "
<< dato << std::endl;
       // if(dato == '2') return; /*Error
1, cuando se para cuando está cuadrando,
debe salir de la funcion, el Arduino se r
establece*/
        //Cuando la camara tiene compensa
cion automatica de exposicion capturamos
siete imagenes hasta que se estabilice
        imagen.release();
        for (int k = 0; k < 7; k++) {
            cap >> imagen; //Capturamos u
na imagen
        }
        preprocesamiento();
    }
    std::cout << "t" << std::endl;
    serialPutchar(fd, '9');
    while (!serialDataAvail(fd)) {
    } // Espero hasta que haya datos en e
l puerto serial
    char dato = serialGetchar(fd); //Vari
able para que se transfiera del buffer el
dato
    while (dato != '2') {
```

```
        serialPutchar(fd, 't');
        while (!serialDataAvail(fd)) {
        } // Espero hasta que haya datos en el puerto serial
        dato = serialGetchar(fd); //Variable para que se transfiera del buffer el dato
        std::cout << "Dato bucle while envio t: " << dato << std::endl;
    }
    std::cout << "Finalizo centrado" << dato << std::endl;
    std::cout << "Posicion y: " << rectaobjetivo.y << " Posicion x: " << rectaobjetivo.x << std::endl;
    std::cout << " Posicion otro costado: " << rectaobjetivo.y + rectaobjetivo.height << " alto " << imagen.rows << std::endl;
#endif
    /* //Proceso la imagen en la red para saber si es un engranaje
     float probabilidadengrane = clasificar();
     std::cout << "Probabilidad engrane: " << probabilidadengrane << std::endl;
     if (probabilidadengrane > 0.2) {
         std::cout << "Es un engrane! Sigue el proceso" << std::endl;
     } else {
         std::cout << "La imagen no pertenece a un engrane, abortando proceso" << std::endl;
         //errores += "No es engrane!";
         //return false;
     }*/
    imagencontours = cv::Mat(imagen.rows, imagen.cols, CV_8UC3, cv::Scalar(0, 0, 0));
    roi = cv::Mat(imagen.rows, imagen.cols, CV_8UC3, cv::Scalar(0));
    cv::drawContours(imagencontours, contours, indicemayor, cv::Scalar(0, 255, 0), 1);
    //Buscar minima distancia entre contornos
    cv::Point2f direccion;

    struct puntosmenores {
        int distancia = 1000;
        cv::Point punto = cv::Point(0, 0);
    };
    std::vector<puntosmenores> indicespuntos(6);
    float angulo;
    //Busco la minima distancia cada 60º para sacar el diametro interior del engrane
    for (int j = 0; j < contours[indicemayor].size(); j++) {
        direccion = cv::Point2f(contours.at(indicemayor).at(j).x, contours.at(indicemayor).at(j).y) - centrocirculo;
        angulo = atan2f((float) direccion.y, (float) direccion.x);
        distanciacentropunto = cv::norm(cv::Point2f(contours.at(indicemayor).at(j).x, contours.at(indicemayor).at(j).y) - centrocirculo);
        if (angulo >= 0 && angulo < (M_PI / 3)) {
            if (distanciacentropunto < indicespuntos.at(0).distancia) {
                indicespuntos.at(0).distancia = distanciacentropunto;
                indicespuntos.at(0).punto = contours.at(indicemayor).at(j);
            }
        } else if (angulo >= (M_PI / 3) && angulo < (2 * M_PI / 3)) {
            if (distanciacentropunto < indicespuntos.at(1).distancia) {
                indicespuntos.at(1).distancia = distanciacentropunto;
                indicespuntos.at(1).punto = contours.at(indicemayor).at(j);
            }
        } else if (angulo >= (2 * M_PI / 3) && angulo <= M_PI) {
            if (distanciacentropunto < indicespuntos.at(2).distancia) {
                indicespuntos.at(2).distancia = distanciacentropunto;
                indicespuntos.at(2).punto = contours.at(indicemayor).at(j);
            }
        } else if (angulo >= -M_PI && angulo < -(2 * M_PI / 3)) {
            if (distanciacentropunto < indicespuntos.at(3).distancia) {
                indicespuntos.at(3).distancia = distanciacentropunto;
                indicespuntos.at(3).punto = contours.at(indicemayor).at(j);
            }
        } else if (angulo >= -2 * M_PI / 3 && angulo < -M_PI / 3) {
            if (distanciacentropunto < indicespuntos.at(4).distancia) {
                indicespuntos.at(4).distancia = distanciacentropunto;
                indicespuntos.at(4).punto = contours.at(indicemayor).at(j);
            }
        } else if (angulo >= -M_PI / 3 && angulo < 0) {
            if (distanciacentropunto < indicespuntos.at(5).distancia) {
                indicespuntos.at(5).distancia = distanciacentropunto;
                indicespuntos.at(5).punto = contours.at(indicemayor).at(j);
            }
        }
    }
    circulointerno.clear();
    //Añado al vector circulo intenro
    for (int j = 0; j < indicespuntos.size(); j++) {
        circulointerno.push_back(indicespuntos.at(j).punto);
```

```
    }
    minEnclosingCircle(circulointerno, ce
ntrocirculointerno, radiocirculointerno);
//Saco el circulo interno que entra con e
sos puntos

    bool iniciocadena = false;
    bool estado = false;
    //Calculo el radio intermedio para se
parar entre cresas y valles
    float radiointermedio = radiocirculoi
nterno + (radioexterior - radiocirculoint
erno) / 2;
    int maximovector = contours[indicemay
or].size();
    //Añado las cadenas de contornos sepa
randolas si son crestas o valles
    for (int i = 0; i < maximovector; i++
) {
        distanciacentropunto = cv::norm(c
v::Point2f(contours.at(indicemayor).at(i)
.x, contours.at(indicemayor).at(i).y) - c
entrocirculo);
        if (distanciacentropunto > radioi
ntermedio) estado = true;
        else estado = false;
        if (!iniciocadena) {
            if (estado) {
         int iterador = maximovector - 1;
                distanciacentropunto = cv
::norm(cv::Point2f(contours.at(indicemayo
r).at(iterador).x, contours.at(indicemayo
r).at(iterador).y) - centrocirculo);
                while (distanciacentropun
to > radiointermedio) {
                    distanciacentropunto
= cv::norm(cv::Point2f(contours.at(indice
mayor).at(iterador).x, contours.at(indice
mayor).at(iterador).y) - centrocirculo);
                    iterador -= 1;
                }
                for (int j = iterador; j
< maximovector; j++) {
                    cresta.push_back(cont
ours.at(indicemayor).at(j));
                }
                cresta.push_back(contours
.at(indicemayor).at(i));
                maximovector = iterador +
1;
                iniciocadena = true;
            } else {
                int iterador = maximovect
or - 1;
                distanciacentropunto = cv
::norm(cv::Point2f(contours.at(indicemayo
r).at(iterador).x, contours.at(indicemayo
r).at(iterador).y) - centrocirculo);
                while (distanciacentropun
to < radiointermedio) {
                    distanciacentropunto
= cv::norm(cv::Point2f(contours.at(indice
mayor).at(iterador).x, contours.at(indice
mayor).at(iterador).y) - centrocirculo);
                    iterador -= 1;
                }
                for (int j = iterador; j
< maximovector; j++) {
                    valle.push_back(conto
urs.at(indicemayor).at(j));
                }
                valle.push_back(contours.
at(indicemayor).at(i));
                maximovector = iterador +
1;
                iniciocadena = true;
            }
        } else if (estado && iniciocadena
) {
            if (valle.size() > 0) {
                //Verifica si el punto qu
e se añade es demasiado pequeño para cons
iderarse diente
                if (cv::contourArea(valle
) > 2) {
                    vallestotales.push_ba
ck(valle);
                }
                valle.clear();
            }
            cresta.push_back(contours.at(
indicemayor).at(i));
        } else if (!estado && iniciocaden
a) {
            if (cresta.size() > 0) {
                if (cv::contourArea(crest
a) > 2) {
                    crestastotales.push_b
ack(cresta);
                }
                cresta.clear();
            }
            valle.push_back(contours.at(i
ndicemayor).at(i));

        }
    }
    if (valle.size()) {
        vallestotales.push_back(valle);
        valle.clear();
    }
    if (cresta.size()) {
        crestastotales.push_back(cresta);
        cresta.clear();
    }

    numerocrestas = crestastotales.size()
;
    numerovalles = vallestotales.size();

    std::vector<areasrectas> areascrestas
, areasvalles;
    cv::Rect rectaerrores;
    contours_poly.resize(numerocrestas);
    areasrectas estructuracrestas, estruc
turavalles;
    cv::Moments momentoscrestas;
    //Calculo las areas de las crestas y
el centro de masa de las mimsmas
    for (int i = 0; i < numerocrestas; i+
+) {
```

```
        cv::drawContours(lineasexternas,
crestastotales, i, cv::Scalar(244, 200, 2
54), 1, cv::LINE_AA);
        areacresta = cv::contourArea(cres
tastotales.at(i));
        areapromediocrestas += areacresta
;
        estructuracrestas.area = areacres
ta;
        approxPolyDP(crestastotales[i], c
ontours_poly[i], 3, true);
        estructuracrestas.Recta = cv::bou
ndingRect(contours_poly[i]);
        momentoscrestas = cv::moments(cre
stastotales.at(i), true);
        estructuracrestas.centroide = cv:
:Point2f((momentoscrestas.m10 / momentosc
restas.m00), (momentoscrestas.m01 / momen
toscrestas.m00));
        cv::circle(lineasexternas, estruc
turacrestas.centroide, 1, cv::Scalar(0, 1
25, 255), -1);
        areascrestas.push_back(estructura
crestas);
    }
    areapromediocrestas /= (double) numer
ocrestas;
    contours_poly.resize(numerovalles);
    //Calculo el area de los valles y los
cenetroides de las mismas
    for (int i = 0; i < numerovalles; i++
) {
        cv::drawContours(lineasinternas,
vallestotales, i, cv::Scalar(244, 255, 20
4), 1, cv::LINE_AA);
        areavalle = cv::contourArea(valle
stotales.at(i));
        areapromediovalles += areavalle;
        estructuravalles.area = areavalle
;
        approxPolyDP(vallestotales[i], co
ntours_poly[i], 3, true);
        estructuravalles.Recta = cv::boun
dingRect(contours_poly[i]);
        areasvalles.push_back(estructurav
alles);

    }
    areapromediovalles /= (double) numero
valles;
    //Sacar la desviacion estandar de las
crestas
    for (int i = 0; i < numerocrestas; i+
+) {
        desviacionestandarcresta += powf(
(areapromediocrestas - areascrestas.at(i)
.area), 2);
    }
    desviacionestandarcresta = sqrtf(desv
iacionestandarcresta / numerocrestas);
    for (int i = 0; i < numerovalles; i++
) {
        desviacionestandarvalle += powf((
areapromediovalles - areasvalles.at(i).ar
ea), 2);
    }
    desviacionestandarvalle = sqrtf(desvi
acionestandarvalle / numerovalles);
    std::cout << "Desviacion estandar are
as de crestas: " << desviacionestandarcre
sta \
            << " Area de crestas " << are
apromediocrestas << " nie - 2*de " \
            << areapromediocrestas - 2 *
desviacionestandarcresta \
            << " niu + 2*de " << areaprom
ediocrestas + 2 * desviacionestandarcrest
a << std::endl;
    std::cout << "Desviacion estandar are
as de valles: " << desviacionestandarvall
e \
            << " Area de crestas " << are
apromediovalles << " nie - 2*de " \
            << areapromediovalles - 2 * d
esviacionestandarvalle \
            << " niu + 2*de " << areaprom
ediovalles + 2 * desviacionestandarvalle
<< std::endl;
    //Aplicar la distribucion normal esta
ndar para encontrar areas con dientes par
tidos
    for (int i = 0; i < numerocrestas; i+
+) {
        if (areascrestas.at(i).area < are
apromediocrestas - 2 * desviacionestandar
cresta && areascrestas.at(i).area > areap
romediocrestas + 2 * desviacionestandarcr
esta) {
            numeroerrores += 1;
            errores += "P" + std::to_stri
ng(numeroerrores) + " Diente partido\n";
            cv::rectangle(imagencontours,
areascrestas.at(i).Recta, cv::Scalar(0, 0
, 255));
            cv::putText(imagencontours, "
P" + std::to_string(numeroerrores), cv::P
oint(areascrestas.at(i).Recta.x + areascr
estas.at(i).Recta.width / 2, areascrestas
.at(i).Recta.y + areascrestas.at(i).Recta
.height / 2), cv::FONT_HERSHEY_PLAIN, 1,
cv::Scalar(0, 0, 255));
        }
    }
    //Aplicar la distribucion normal esta
ndar para encontrar areas con acumulacion
de material
    for (int i = 0; i < numerovalles; i++
) {
        if (areasvalles.at(i).area < area
promediovalles - 2 * desviacionestandarva
lle && areasvalles.at(i).area > areaprome
diovalles + 2 * desviacionestandarvalle)
{
            numeroerrores += 1;
            errores += "P" + std::to_stri
ng(numeroerrores) + " Acumulacion materia
l\n";
            cv::rectangle(imagencontours,
areasvalles.at(i).Recta, cv::Scalar(0, 0,
255));
            cv::putText(imagencontours, "
P" + std::to_string(numeroerrores), cv::P
```

```
oint(areasvalles.at(i).Recta.x + areasval
les.at(i).Recta.width / 2, areasvalles.at
(i).Recta.y + areasvalles.at(i).Recta.hei
ght / 2), cv::FONT_HERSHEY_PLAIN, 1, cv::
Scalar(0, 0, 255));
        }
    }
    //Verifica si no hay dientes
    double anguloactual = 0.0, anguloante
rior = 0.0;
    double angulomenor = 360, diferencia
= 0.0;
    for (int i = 0; i < numerocrestas; i+
+) {
        if (i == 0) anguloanterior = atan
2f(centrocirculointerno.y - areascrestas.
at(numerocrestas - 1).centroide.y, areasc
restas.at(numerocrestas - 1).centroide.x
- centrocirculointerno.x);
        else anguloanterior = atan2f(cent
rocirculointerno.y - areascrestas.at(i -
1).centroide.y, areascrestas.at(i - 1).ce
ntroide.x - centrocirculointerno.x);
        anguloactual = atan2f(centrocircu
lointerno.y - areascrestas.at(i).centroid
e.y, areascrestas.at(i).centroide.x - cen
trocirculointerno.x);
        anguloactual = (anguloactual > 0.
0 ? anguloactual : (2 * M_PI + anguloactu
al));
        anguloanterior = (anguloanterior
> 0.0 ? anguloanterior : (2 * M_PI + angu
loanterior));
        diferencia = std::abs(anguloactua
l - anguloanterior)* 180.0 / M_PI;
        if(diferencia > 90) diferencia =
360 - diferencia; //cuando pasa un diente
antes de 360 y el otro despues de cero me
aseguro que no sume toda la vuelta
        areascrestas.at(i).anguloentrecre
stas = diferencia;
    }
    //Saco la maxima frecuencia de angulo
s
    int frecuencia = 0, frecuenciamaxima
= 0;
    float promediofrecuencia = 0.0;
    for (int i = 0; i < numerocrestas; i+
+) {
        frecuencia = 0;
        promediofrecuencia = 0;
        for (int j = 0; j < numerocrestas
; j++) {
            if ((int) areascrestas.at(i).
anguloentrecrestas == (int) areascrestas.
at(j).anguloentrecrestas) {
                frecuencia += 1;
                promediofrecuencia += are
ascrestas.at(i).anguloentrecrestas;
            }
        }
        if (frecuencia > frecuenciamaxima
) {
            frecuenciamaxima = frecuencia
;
            angulomenor = promediofrecuen
cia / (float) frecuencia;
        }
    }
    std::cout << "Angulo con mas frecuenc
ia: " << angulomenor << std::endl;
    float sumaangulos = 0.0;
    float anguloactualgrados = 0.0;
    float anguloactualcorregido = 0.0;
    //Busco desde el inicio de los angulo
s los dientes a traves de la suma del ang
ulo con mas frecuencia
    for (int i = 0; i < numerocrestas; i+
+) {
        if (i == 0) {
            anguloactual = atan2f(centroc
irculointerno.y - areascrestas.at(i).cent
roide.y, areascrestas.at(i).centroide.x -
centrocirculointerno.x);
            anguloactualgrados = 180 * an
guloactual / M_PI;
            anguloactualcorregido = (angu
loactualgrados > 0.0 ? anguloactualgrados
: (360.0 + anguloactualgrados));
            sumaangulos = anguloactualcor
regido;
            sumatotalcircunferencia += ar
eascrestas.at(i).anguloentrecrestas;
        } else {
            sumaangulos += angulomenor;
            if (sumaangulos > 360) sumaan
gulos -= 360;
            anguloactual = atan2f(centroc
irculointerno.y - areascrestas.at(i).cent
roide.y, areascrestas.at(i).centroide.x -
centrocirculointerno.x);
            anguloactualgrados = 180 * an
guloactual / M_PI;
            anguloactualcorregido = (angu
loactualgrados > 0.0 ? anguloactualgrados
: (360.0 + anguloactualgrados));
            if (anguloactualcorregido <=
2 * offset) anguloactualcorregido += 360;
            if (sumaangulos <= 2 * offset
) sumaangulos += 360;
            limitesuperior = sumaangulos
+ offset;
            limiteinferior = sumaangulos
- offset;
            if (anguloactualcorregido <=
limitesuperior && anguloactualcorregido >
= limiteinferior) {
                sumaangulos = anguloactua
lcorregido;
                sumatotalcircunferencia +
= areascrestas.at(i).anguloentrecrestas;
            } else {
                numeroerrores += 1;
                dientesperdidos += 1;
                cv::Point2f puntoerror =
centrocirculo + cv::Point2f(radiointermed
io * cosf(sumaangulos * M_PI / 180.0), -r
adiointermedio * sinf(sumaangulos * M_PI
/ 180.0));
```

```
                std::cout << "Punto error
: [" << puntoerror.x << "," << puntoerror
.y << "]" << std::endl;
                errores += "P" + std::to_
string(numeroerrores) + " No hay diente\n
";
                cv::circle(imagencontours
, puntoerror, 5, cv::Scalar(0, 0, 255), -
1);
                cv::putText(imagencontour
s, "P" + std::to_string(numeroerrores), p
untoerror, cv::FONT_HERSHEY_PLAIN, 1.5, c
v::Scalar(0, 0, 255));
                std::cout << "\033[1;31m
No hay diente en esta ubicacion \033[0m"
<< std::endl;
                sumaangulos = anguloactua
lcorregido;
                sumatotalcircunferencia +
= angulomenor;
            }
        }
    }
    if (sumatotalcircunferencia <= 360 -
offset && dientesperdidos == 0) { //Si no
hay dientes perdidos reviso que la circun
ferencia total cumpla los 360º
        sumaangulos += angulomenor;
        numeroerrores += 1;
        dientesperdidos += 1;
        cv::Point2f puntoerror = centroci
rculo + cv::Point2f(radiointermedio * cos
f(sumaangulos * M_PI / 180.0), -radiointe
rmedio * sinf(sumaangulos * M_PI / 180.0)
);
        std::cout << "Punto error: [" <<
puntoerror.x << "," << puntoerror.y << "]
" << std::endl;
        errores += "P" + std::to_string(n
umeroerrores) + " No hay diente\n";
        cv::circle(imagencontours, puntoe
rror, 5, cv::Scalar(0, 0, 255), -1);
        cv::putText(imagencontours, "P" +
std::to_string(numeroerrores), puntoerror
, cv::FONT_HERSHEY_PLAIN, 1.5, cv::Scalar
(0, 0, 255));
        std::cout << "\033[1;31m No hay d
iente en esta ubicacion \033[0m" << std::
endl;
    }
    modulo = (2 * radioexterior * constan
teconversionpxacm) / (crestastotales.size
() + dientesperdidos + 2); //M = (De)/(Z
+ 2)
    if ((modulo - (int) modulo) <= 0.7 &&
(modulo - (int) modulo) >= 0.3) {
        modulo = (int) modulo + 0.5;
    } else if ((modulo - (int) modulo) >
0.7) {
        modulo = (int) modulo + 1.0;
    } else if ((modulo - (int) modulo) <
0.3) {
        modulo = (int) modulo + 0.0;
    }
    paso = modulo * M_PI;
    roi = imagencontours(rectaobjetivo);
    std::cout << "*****Informacion del en
grane*****" << std::endl;
    std::cout << "Hay " << dientesperdido
s << " dientes perdidos" << std::endl;
    std::cout << "* Diametro exterior(De)
: " << 2 * radioexterior * constanteconve
rsionpxacm << " mm" << std::endl;
    std::cout << "* Diametro interior(Di)
: " << 2 * radiocirculointerno * constant
econversionpxacm << " mm" << std::endl;
    std::cout << "* Modulo(M): " << modul
o << std::endl;
    std::cout << "* Paso(P): " << paso <<
std::endl;
    std::cout << "* Numero de dientes(Z):
" << crestastotales.size() + dientesperdi
dos << std::endl;
    std::cout << "**********************
**********" << std::endl;
    cv::putText(imagencontours, "Crestas:
" + std::to_string(numerocrestas), centro
circulo - cv::Point2f(20, 10), cv::FONT_H
ERSHEY_PLAIN, 1, cv::Scalar(0, 125, 255))
;
    cv::putText(imagencontours, "Valles:
" + std::to_string(numerovalles), centroc
irculo - cv::Point2f(20, -10), cv::FONT_H
ERSHEY_PLAIN, 1, cv::Scalar(125, 0, 255))
;
    cv::circle(imagencontours, centrocirc
ulo, radioexterior, cv::Scalar(255, 100,
100));
    cv::line(imagencontours, centrocircul
o - cv::Point2f(20, 30), centrocirculo -
cv::Point2f(-25, 30), cv::Scalar(255, 100
, 100), 3);
    cv::putText(imagencontours, "D.E.", c
entrocirculo - cv::Point2f(45, 30), cv::F
ONT_HERSHEY_PLAIN, 1, cv::Scalar(255, 100
, 100));
    cv::circle(imagencontours, centrocirc
ulointerno, radiocirculointerno, cv::Scal
ar(255, 255, 255));
    cv::line(imagencontours, centrocircul
o - cv::Point2f(15, 50), centrocirculo -
cv::Point2f(-25, 50), cv::Scalar(255, 255
, 255), 3);
    cv::putText(imagencontours, "D.I.", c
entrocirculo - cv::Point2f(45, 50), cv::F
ONT_HERSHEY_PLAIN, 1, cv::Scalar(255, 255
, 255));
    cv::circle(imagencontours, centrocirc
ulo, (int) radiointermedio, cv::Scalar(12
5, 255, 255));
    cv::line(imagencontours, centrocircul
o - cv::Point2f(15, 70), centrocirculo -
cv::Point2f(-25, 70), cv::Scalar(125, 255
, 255), 3);
    cv::putText(imagencontours, "D.P.", c
entrocirculo - cv::Point2f(45, 70), cv::F
ONT_HERSHEY_PLAIN, 1, cv::Scalar(125, 255
, 255));
    std::cout << errores << std::endl;
#ifdef mostrarcapturas
    cv::namedWindow("ENGRANAJE ORIGINAL",
cv::WINDOW_NORMAL);
```

```
    cv::imshow("ENGRANAJE ORIGINAL", imag
en);
    cv::imshow("Lineas externas", linease
xternas);
    cv::imshow("Lineas internas", lineasi
nternas);
    cv::imshow("GEAR VERSION IMAGEN GRIS
INVERSA: OTSU: ", imagengris);
    cv::imshow("Contorno del engranaje",
imagencontours);
    cv::namedWindow("ROI", cv::WINDOW_NOR
MAL);
    cv::imshow("ROI", roi);
    cv::waitKey();
    cv::destroyAllWindows();
#endif
    bool erroresfinales = false;
    if (errores.compare("") == 0) errores
finales = true;
    else erroresfinales = false;
    return erroresfinales;
}
#endif /* VISION_H */
```

ANEXO 7 – PROGRAMACIÓN ARDUINO.

```
#include "ArduinoJson.h" //Generador
de Json para C++, el asistente se pue
de encontrar aca https://arduinojson.
org/v5/assistant/
//LED MULTICOLOR
int red = 6;
int blue = 3;
int green = 5;
int objbueno = 13;
int objmalo = 8;
int datoumbral = 0;
// INICIO Y PARO
int emerg = 9;
int start = 7;
int ledcamara = 4;

//Variables para iniciar proceso y pa
rar el proceso
bool inicioproceso = false;
bool paroemergencia = false;
bool modofuncionamiento = true;
int datosentidomotor = 0;
int datotoclasificar = 0;
int datoestado = 0;
bool stanby = false;
bool iluminacion = false;
//Variables LED rgb
int rgb0 = 255;   //Rojo
int rgb1 = 255;   //Verdel
int rgb2 = 255;   //Azul
bool sis_activado = false;
bool sis_activado2 = false;
bool resetdatosentido = false;
int receptor = 0;
int umbral = 450;
#include <Servo.h>
Servo clasificador;
int pinservo = 10;
#define motorpwm 11
int motordir = 12;
bool actd = false;
bool antd = false;
bool vidprocesado = false;
bool clasificadorservo = false;
int dato = 70;
int datoclasificacion = 80;
int valorinicial = 0; //Valor de la e
ntrada infrarrojo
int valor = 0; //Valor de la entrada
infrarrojo
int dispositivo = 0; //Identificador
dispositivo
char c, d;
bool centrado = false;
//Variables para Arduino JSON Generat
or/Parser
const size_t bufferSize =  JSON_OBJEC
T_SIZE(11);
DynamicJsonBuffer Buffersalida(buffer
Size);
JsonObject& datosalidajson = Buffersa
lida.createObject();
unsigned long tiempoinicio = 0;
String datospuertoserial = "";
bool haycadena = false;
bool iniciocadena = false;
bool fincadena = false;
int estado = 0;
bool iniciarmanual = false; //Inicia
una vez el modo manual
#define debug 1
void setup() {
  Serial.begin(115200);
  pinMode(red, OUTPUT);
  pinMode(blue, OUTPUT);
  pinMode(green, OUTPUT);
  pinMode(objmalo, OUTPUT);
  pinMode(objbueno, OUTPUT);
  pinMode(motorpwm, OUTPUT);
  pinMode(motordir, OUTPUT);
  pinMode(emerg, INPUT);
  pinMode(start, INPUT);
  pinMode(ledcamara, OUTPUT);
  digitalWrite(motordir, HIGH);
  clasificador.attach(pinservo);
  clasificador.write(110);
  analogWrite(motorpwm, 0);
  analogWrite(red, 255 - rgb0);
  analogWrite(blue, 255 - rgb1);
  analogWrite(green, 255 - rgb2);
  valor = analogRead(receptor);
  Serial.setTimeout(1000);
  /*solo para manual de momento*/
  if ( (valorinicial >= 0) || (valori
nicial <= 1000)) {
    sis_activado = false;
  } else {
    sis_activado = true;
  }
  if ( valor <= umbral) {
    actd = true;
  } else {
    actd = false;
  }
  datosalidajson["presencia"] = actd;
  if (digitalRead(emerg) == LOW) {
    paroemergencia = true;
  }
  else paroemergencia = false;
  if (paroemergencia == true) {
    estado = -1;
    datosalidajson["estado"] = -1;
```

```
  }  else {
    estado = 0;
    datosalidajson["estado"] = 0;
  }
  antd = actd;
  datosalidajson.printTo(Serial);
}
void loop() {
 if (Serial.available() > 0) {
   //  Serial.print("Datos serial:");
Serial.print(Serial.available());
  }
while (Serial.available() > 0) { //Si
empre leo los datos del Serial
  haycadena = true;
  d = Serial.read();
   //Verifico que me lleguen las llav
es { y } antes de procesar el objeto
 if (d == '{') {
      datospuertoserial = "";
      iniciocadena = true;
  }
  if (iniciocadena) {
     datospuertoserial.concat(d);
     rgb0 = 20, rgb1 = 20, rgb2 = 20;
      analogWrite(red, 255 - rgb0);
      analogWrite(blue, 255 - rgb1);
      analogWrite(green, 255 - rgb2);
    }
    if (d == '}') {
      fincadena = true;
      break;
    }
  } ////Fin del Serial Available
 if (iniciocadena && fincadena) {
    iniciocadena = false, fincadena =
false;
   rgb0 = 255, rgb1 = 255, rgb2 = 0;
    analogWrite(red, 255 - rgb0);
    analogWrite(blue, 255 - rgb1);
    analogWrite(green, 255 - rgb2);
    delay(200);
 DynamicJsonBuffer Bufferentrada(buff
erSize);
 JsonObject& datojson = Bufferentrada
.parseObject(datospuertoserial);
    if (datojson.success()) {
    rgb0 = 255, rgb1 = 255, rgb2 = 0;
      analogWrite(red, 255 - rgb0);
      analogWrite(blue, 255 - rgb1);
      analogWrite(green, 255 - rgb2);
      delay(200);
      for (auto kv : datojson) {
        if (String(kv.key).compareTo(
"dispositivo") == 0) {
  dispositivo = kv.value.as<unsigned
int>();
       }
    else if (String(kv.key).compareTo
("modofunc") == 0) {
    modofuncionamiento = kv.value.as<
bool>();
    if (modofuncionamiento) Serial.pr
int("auto");
     else {
            iniciarmanual = true;
            Serial.println("manual");
        }
      }
    else if (String(kv.key).compareTo
("motor") == 0) {
    dato = kv.value.as<signed int>();
    datoclasificacion = dato;
    }
    else if (String(kv.key).compareTo
("estado") == 0) {
     datoestado = kv.value.as<signed
int>();
    if (datoestado == -1) {
      paroemergencia = true;
     inicioproceso = false;
     rgb0 = 120, rgb1 = 0, rgb2 = 0;
      analogWrite(motorpwm, 0);
      if (estado != -1) {
      estado = -1;
       datosalidajson["estado"] = -1;
      datosalidajson.printTo(Serial);
       }
      }
      else if (datoestado == 0) {
       paroemergencia = false;
       inicioproceso = false;
       centrado = false;
     rgb0 = 255, rgb1 = 69, rgb2 = 0;
//Naranja
      analogWrite(motorpwm, dato);
      if (estado != 0) {
         estado = 0;
        datosalidajson["estado"] = 0;
       datosalidajson.printTo(Serial);
        }
       }
     else if (datoestado == 1) {
          if (estado != 1) {
             estado = 1;
      datosalidajson["estado"] = 1;
     datosalidajson.printTo(Serial);
            }
     centrado = false;
      paroemergencia = false;
       inicioproceso = true;
       dato = datoclasificacion;
      rgb0 = 0, rgb1 = 100, rgb2 = 0;
//Verde
      }
    }
```

```
    else if (String(kv.key).compareTo
("clasificador") == 0) {
 clasificadorservo = kv.value.as<bool
ean>();
        }
  else if (String(kv.key).compareTo("
imagenprocesada") == 0) {
  rgb0 = 255, rgb1 = 255, rgb2 = 255;
    analogWrite(red, 255 - rgb0);
    analogWrite(blue, 255 - rgb1);
    analogWrite(green, 255 - rgb2);
    vidprocesado = kv.value.as<boolea
n>();
    Serial.print("vidprocesado:"); Se
rial.print(vidprocesado);
        }
    else if (String(kv.key).compareTo
("sentidomotor") == 0  ) {
          datosentidomotor = kv.value
.as<signed int>();
        }
        else if (String(kv.key).compa
reTo("direcclasificacion") == 0  ) {
          datotoclasificar = kv.value
.as<signed int>();
        }
        else if (String(kv.key).compa
reTo("umbral") == 0) {
          datoumbral = kv.value.as<si
gned int>();
        }
        else if (String(kv.key).compa
reTo("capturarengrane") == 0) {
          iluminacion = kv.value.as<b
oolean>();
        }
      }//Fin del For
      datospuertoserial = "";
    }//Fin del procesamiento de JSOn
    else {
      Serial.print("Falla JSON"); Ser
ial.println(datospuertoserial); //Si
por algun motivo no procesa bien, dev
uelva al dispositivo que genero la se
ñal
    }
    datospuertoserial = "";
  }
  //Fin de procesamiento de JSON
  if (digitalRead(emerg) == LOW) {
    centrado = false;
    rgb0 = 255, rgb1 = 0, rgb2 = 0; /
/ROJO
    analogWrite(red, 255 - rgb0);
    analogWrite(green, 255 - rgb1);
    analogWrite(blue, 255 - rgb2);

    inicioproceso = false;
    paroemergencia = true;
  }
  /*else if (paroemergencia == true &
& digitalRead(emerg) == HIGH) {
    paroemergencia = true;
    }*/
  else {
    //Serial.print("emergenciaoff");
    paroemergencia = false;
  }
  if (digitalRead(start) == HIGH) {
    delay(500);
    if ( digitalRead(start) == HIGH)
{
      rgb0 = 255, rgb1 = 69, rgb2 = 0
; //Naranja
      inicioproceso = false;
      centrado = false;
      datosalidajson["estado"] = 0;
      delay(1000);
    }
    else {
      delay(500);
      inicioproceso = true;
      dato = 80;
      datosalidajson["estado"] = 1;
      rgb0 = 0, rgb1 = 100, rgb2 = 0;
    }
  }

  if (paroemergencia) {
    //Serial.print("emergencia");
    analogWrite(motorpwm, 0);
    inicioproceso = false;
    if (estado != -1) {
      estado  = -1;
      datosalidajson["estado"] = -1;
      datosalidajson.printTo(Serial);
    }
    //
aqui añado
    analogWrite(red, 255 - rgb0);
    analogWrite(green, 255 - rgb1);
    analogWrite(blue, 255 - rgb2);
    // paroemergencia = false; //aqui
añado
  }
  else { //Else Emergencia
    ///Serial.print("elseparoemergenc
ia");
    if (inicioproceso && estado != 1
&& !paroemergencia ) {
      Serial.print("run");
      estado = 1;
      datosalidajson["estado"] = 1;
      datosalidajson.printTo(Serial);
    }
    else if (!inicioproceso && estado
!= 0 && !paroemergencia  ) {
      Serial.print("stop");
```

```
      rgb0 = 255, rgb1 = 69, rgb2 = 0
; //Naranja
      estado = 0;
      datosalidajson["estado"] = 0;
      datosalidajson.printTo(Serial);

    }
    if (digitalRead(start) == HIGH) {
      delay(500);
      if ( digitalRead(start) == HIGH
) {
        rgb0 = 255, rgb1 = 69, rgb2 =
0; //Naranja
        inicioproceso = false;
        centrado = false;
        datosalidajson["estado"] = 0;
        delay(1000);
      }
      else {
        delay(500);
        inicioproceso = true;
        dato = 80;
        datosalidajson["estado"] = 1;
      rgb0 = 0, rgb1 = 100, rgb2 = 0;
      }
    }
  }
  if (modofuncionamiento == true) {
    //Serial.print("a");
    if (inicioproceso) {
      Serial.print("i");
      digitalWrite(ledcamara, LOW);
      analogWrite(red, 255 - rgb0);
      analogWrite(green, 255 - rgb1);
      analogWrite(blue, 255 - rgb2);
      analogWrite(motorpwm, dato);
      valor = analogRead(receptor);
      if ( valor <= umbral) {
        actd = true;
      }
      else {
        actd = false;
      }
      delay(300);
   if ( actd != antd) {
  datosalidajson["presencia"] = actd;
   if (inicioproceso) {
      datosalidajson["estado"] = 1;
    }
   else datosalidajson["estado"] = 0;
     datosalidajson.printTo(Serial);
        delay(100);
    }
   if (actd == true && vidprocesado =
= false) {
       digitalWrite(ledcamara, HIGH);
        dato = 0;
        analogWrite(motorpwm, dato);
        if (!centrado) {
          Serial.print("c");
          centrado = true;
          c = '0';
          tiempoinicio = millis();

    while ( c != '9' ) {
    if (millis() - tiempoinicio > 100
00) break;
     if (Serial.available()) {
          c = Serial.read();
        if (c == 'a') {
        digitalWrite(motordir, HIGH);
         analogWrite(motorpwm, 80);
    rgb0 = 0, rgb1 = 255, rgb2 = 255;
    analogWrite(red, 255 - rgb0);
    analogWrite(green, 255 - rgb1);
    analogWrite(blue, 255 - rgb2);
     delay(100);
      Serial.print('1');
      }
        else if (c == 'd') {
  rgb0 = 125, rgb1 = 125, rgb2 = 125;
   analogWrite(red, 255 - rgb0);
   analogWrite(green, 255 - rgb1);
   analogWrite(blue, 255 - rgb2);
   digitalWrite(motordir, LOW);
  analogWrite(motorpwm, 80);
   delay(100);
   Serial.print('1');
        }
   else if (c == '2') break;
   analogWrite(motorpwm, 0);
   digitalWrite(motordir, HIGH);
     delay(5);
      }
if (millis() - tiempoinicio > 10000)
break;
   rgb0 = 0, rgb1 = 100, rgb2 = 100;
  analogWrite(red, 255 - rgb0);
   analogWrite(green, 255 - rgb1);
   analogWrite(blue, 255 - rgb2);
  if (digitalRead(emerg) == LOW) {
      centrado = false;
       paroemergencia = true;
    if (estado != -1) {
         break;
       estado = -1;
       datosalidajson["estado"] = -1;
    datosalidajson.printTo(Serial);
      } break;
   }
  if (digitalRead(start) == HIGH) {
              delay(500);
              if ( digitalRead(start)
== HIGH) {
          Serial.println("R6vt");
     rgb0 = 255, rgb1 = 69, rgb2 = 0;
//Naranja
   inicioproceso = false;
```

```
    centrado = false;
     datosalidajson["estado"] = 0;
       delay(1000);
         break;
        }
        else {
       delay(500);
     modofuncionamiento = true;
     inicioproceso = true;
     dato = 80;
     datosalidajson["estado"] = 1;
     rgb0 = 0, rgb1 = 100, rgb2 = 0;
       }
        break;
      }
    }
     Serial.print('2');
     rgb0 = 0, rgb1 = 255, rgb2 = 0;
     analogWrite(red, 255 - rgb0);
     analogWrite(green, 255 - rgb1);
     analogWrite(blue, 255 - rgb2);
          c = 'f';
        }
      }
    else if (vidprocesado == true)  {
        Serial.println("imagenprocesa
do");
        vidprocesado = false;
        centrado = false;
        rgb0 = 255, rgb1 = 125, rgb2
= 125;
        analogWrite(red, 255 - rgb0);
        analogWrite(green, 255 - rgb1
);
        analogWrite(blue, 255 - rgb2)
;
        dato = datoclasificacion;
        if (clasificadorservo) {
          clasificador.write(110);
          digitalWrite(objbueno, HIGH
);
          digitalWrite(objmalo, LOW);
        }
        else {
          clasificador.write(30);
          digitalWrite(objbueno, LOW)
;
          digitalWrite(objmalo, HIGH)
;
        }

        if (actd == true) {
          valor = analogRead(receptor
);
          if ( valor <= umbral) {
            actd = true;
          }
          else {
            actd = false;
          }
          digitalWrite(ledcamara, LOW);
          analogWrite(motorpwm, dato);
          digitalWrite(motordir, HIGH);
            delay(1000);
         }
        rgb0 = 0, rgb1 = 255, rgb2 = 0;
        }
        antd = actd;
      } else {
        digitalWrite(ledcamara, LOW);
        analogWrite(motorpwm, 0);
        analogWrite(red, 255 - rgb0);
        analogWrite(green, 255 - rgb1);
        analogWrite(blue, 255 - rgb2);
      }
    }
   else {  //  Aqui Hago el modo manual
      rgb0 = 255, rgb1 = 60, rgb2 = 40;
//gris
      analogWrite(red, 255 - rgb0);
      analogWrite(green, 255 - rgb1);
      analogWrite(blue, 255 - rgb2);
      if (iniciarmanual) {
      iniciarmanual = false; //Sole eje
cuta una vez
       Serial.println("MODO MANUAL: ");
       if (!sis_activado) {
         if (datoumbral >= 400) {
           sis_activado2 = true;
         } else sis_activado2 = false;
        }
       if (sis_activado2 == true) {
          digitalWrite(ledcamara, LOW);
          analogWrite(motorpwm, 0);
          tiempoinicio = millis();
          if (datosentidomotor == 1) {
          Serial.println("ADELANTE: ");
           analogWrite(motorpwm, dato);
          digitalWrite(motordir, HIGH);
            delay(1000);
          analogWrite(motorpwm, 0);
       } else analogWrite(motorpwm, 0);
      if (datosentidomotor == -1) {
         Serial.println("ATRAS: ");
         analogWrite(motorpwm, dato);
         digitalWrite(motordir, LOW);
         delay(1000);
           analogWrite(motorpwm, 0);
       } else analogWrite(motorpwm, 0);
        if (iluminacion == true) {
        rgb0 = 20, rgb1 = 55, rgb2 = 0;
         iluminacion = false;
 Serial.println("camaraencendida: ");
       digitalWrite(ledcamara, HIGH);
           delay(2000);
      digitalWrite(ledcamara, LOW);
        } else digitalWrite(ledcamara,
LOW);
```

```
     if ((datotoclasificar == 1) || (
datotoclasificar == 1 && datosentidom
otor == 0) ) {
       digitalWrite(objbueno, HIGH);
       digitalWrite(objmalo, LOW);
       clasificador.write(110);
       Serial.println("bueno");
        }
      else if (datotoclasificar == -1
|| (datotoclasificar == -1 && datosen
tidomotor == 0))  {
        digitalWrite(objmalo, HIGH);
        digitalWrite(objbueno, LOW);
          clasificador.write(30);
          Serial.println("malo");
  } else if (datotoclasificar == 0) {
        }
```

```
         sis_activado == false;
         datosentidomotor = 0;
         datotoclasificar = 0;
           }
       else {
         digitalWrite(ledcamara, LOW);
         analogWrite(motorpwm, 0);
         analogWrite(red, 255 - rgb0);
         analogWrite(green, 255 - rgb1
);
        analogWrite(blue, 255 - rgb2);
       }
    }
  }
}
```

ANEXO 6 – PLANIFICACIÓN DE TAREAS:

N°	ACTIVIDAD	JUL				AGO				SEPT				OCT				NOV				DIC	
		s1	s2	s3	s4	s1	s2	s3	s4	s1	s2	s3	s4	s1	s2	s3	s4	s1	s2	s3	s4	s1	s2
1	Investigación CPS y IoT																						
2	Introducción																						
3	Objetivos																						
4	Estado del Arte																						
5	Marco teórico																						
6	Desarrollo																						
6.1	Elaboración del prototipo																						
6.2	Instalación de Software en Raspberry pi																						
6.3	Desarrollo algoritmo de Vision																						
6.4	Desarrollo algoritmo de cliente -servidor																						
6.5	Desarrollo de interfaz web																						
6.6	Desarrollo algoritmo en el Arduino.																						
7	Pruebas y Resultados																						
8	Elaboración Guía Práctica																						
9	Conclusiones																						
10	Líneas Futuras																						
11	Reporte final y presentación																						